

NORTHWESTERN UNIVERSITY

AutoPoint: Selecting and Interacting with Distant Objects While Traveling in an Automobile

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Amartya Banerjee

EVANSTON, ILLINOIS

July 2018

ABSTRACT

AutoPoint: Selecting and Interacting with Distant Objects While Traveling in an Automobile

Amartya Banerjee

In the near future, self-driving or driverless vehicles will operate without human control, enabling passengers to use their time in new ways. This opens up avenues for designing new interactions and experiences for individuals or groups traveling in an automobile. For that scenario, automobile manufacturers propose developing bigger and better versions of contemporaneous in-car entertainment systems, e.g. larger screens playing movies [6]. This kind of entertainment implicitly seeks to draw our attention away from the outside world that cars are traveling through. I propose an alternative; that interactive car windows are a feasible way to engage with the world one is driving through.

With AutoPoint, I have designed and developed techniques that use a car's window as an interactive and transparent display surface. AutoPoint enables passengers to select and engage with Points-of-Interest (PoI) outside the car. In order to select a PoI, a passenger first takes a snapshot of the area the car is driving through, as seen from her/his point of view, by tapping or pointing at the car window. Thereafter, the passenger can narrow down that snapshot of an area of interest into a particular PoI using multi-touch input. Once the PoI is selected, contextual information—e.g. data about a historic building — is provided on the car window as well.

The system was designed, developed, and studied in the lab (with 36 participants) over multiple iterations. Upon completion of the lab study, I developed a version of AutoPoint as an iPad application. This application incorporated a subset of interaction techniques from the lab setup that were shown to work well. I retrofitted a car with an iPad that mimicked the lab setup as closely

as possible (Figure 5.2) and drove 14 participants on a loop around campus to test the system in the real world and get participant feedback.

In this dissertation, I report on the limitations and successes of multi-touch and in-air gestural inputs when a 2D interactive area is all that is available to observe or learn about an external 3D world in motion. This scenario is obtained, for example, while riding an automobile or a train, when a passenger may view items of interest through the window, and it falls under a class of problems I refer to as Multimodal Transparent Display Interactions (MTDI) with External World in Motion (EWM), or MTDI-EWM for short.

In an age when self-driving cars are being assiduously developed, opportunities exist for a more rewarding touring experience, but that is not the only derivative of this work. The design and evaluation of AutoPoint can also lead to a better appreciation of potential applications in games and informational / educational experiences akin to a museum visit or a guided tour.

Acknowledgements

In terms of things that truly matter in life, I have been dealt a strong hand. I am a lucky guy.

Mike, my advisor and mentor, gave me the freedom to pursue ideas that were of interest to me, work on projects where I was able to collaborate with colleagues from the Learning Sciences, scientists from museums, and school teachers. And on top of that, Mike has always treated me kindly and with respect. I am richer for the experience, and I cannot thank him enough.

Ma and Baba have been instrumental in this journey. I received technical, philosophical, and unflinching moral support from them while I pursued this milestone. More importantly, from encouraging me to play sports, to tinkering with and breaking things around the house, Ma and Baba had let me be a kid :). The importance of which I did not know at the time, but I do now.

Nicole joined me at a crucial stage of the Ph.D. program. Her ability to empathize with me and being kind even in stressful situations was invaluable. Among other things, she knew when a brisk outdoor walk/hike would help, she made sure I did not have to resort to subsisting on graduate student fare and when push came to shove, crunched data that helped me finish on time.

My committee members, Prof. Riesbeck, and Prof. Gergle provided constructive feedback that improved the contribution and quality of my thesis.

My friends and colleagues were an integral part of this experience. Newell introduced me to outdoor rock climbing, but offered much more than that. I treasure the friendship and camaraderie over the years. Nick and Ross gave me the opportunity to start a company together; it was quite the ride and their friendship was the cherry-on-top. Thank you David, Marco, Dan, Kalonji and Zeina for your friendship and advice over the years.

In the wise words of Lester Freamon: "All the pieces matter."

Dedication

For Ma and Baba.

Table of Contents

ABSTRACT	2
ACKNOWLEDGEMENTS	4
DEDICATION	5
LIST OF FIGURES	8
CHAPTER 1 INTRODUCTION	14
1.1 Car travel colors a tourist's impression of a region.	14
1.2 Current and future Automobile systems	16
CHAPTER 2 RELATED WORK	21
2.1 Automotive User Interfaces	22
2.1.1 Car dashboards and interfaces from the past	22
2.1.2 Current in-car infotainment systems and display technology	24
2.1.3 Industrial Design Concepts	27
2.2 In-air gestures, and 3D content navigation via a 2D input device	29
2.2.1 Free-hand pointing and in-air gestures	30
2.3 Navigating 3D Spaces using a 2D input device.	31
2.4 In-Car games and interaction design	33
2.5 Cars as informal learning environments	36
CHAPTER 3 AUTOPOINT PHASE - I	39
3.1 Hardware Design and Assembly	39
3.1.1 Car door mount, projection surface, and projector setup	40
3.1.2 Smart-Tint switching circuit and Hand Tracking System	44
3.2 Software architecture and implementation(s)	47
3.2.1 Pre-pilot software details	48
3.2.2 Testing pre-pilot software	49
3.2.3 Pilot software details	51
3.2.4 Pilot Testing	56
CHAPTER 4 AUTOPOINT PHASE - II	62
4.1 Hardware Refinements	62
4.2 Software changes and refinements	65
4.3 Testing with Participants	71
4.3.1 Test Version 1 (V1)	71

	7
4.3.2 Test Version 2 (V2)	76
CHAPTER 5 AUTOPOINT: PHASE III	85
5.1 Hardware: In-car iPad mount and recorder setup	85
5.2 Software: iPad Application	88
5.2.1 Live View	88
5.2.2 Map View	91
5.3 Test Sessions	92
5.3.1 Participants and testing parameters	92
5.3.2 Participant feedback and discussion	93
CHAPTER 6 CONCLUSION	101
6.1 Problem Domain and Iterative Development	101
6.2 Future Work	103
REFERENCES	106
APPENDIX	115

List of Figures

- Figure 1.1: The Himalayas in India seen from a country road (top), winding roads through Death Valley with geological landmarks (bottom left), and the LA skyline with the San Gabriel mountains in the background (bottom right); all sights and features one can encounter on a road trip..... 15
- Figure 1.2: Ford has outlined its vision of in-car entertainment for self-driving cars where a large projector screen that displays multimedia content is positioned in front of the passengers [6]..... 17
- Figure 1.3: A concept sketch of AutoPoint, where a user interacts with the nearest car window using multi-touch input or in-air gestures detected via a Leap Motion Sensor [48]. 18
- Figure 2.1 (clockwise from top-left): 1923 Ford Model T [64,66] with an ammeter as the only dial on the dashboard; the 1948 Tucker Torpedo [65] with a lot more dials and switches; the 1965 Cadillac Deville [63] with an A/C and radio on the dashboard; the 1988 Nissan Heritage [33] with a car phone and digital displays driven by an onboard computer..... 23
- Figure 2.2: (clockwise from top-left) Apple CarPlay [71], Android Auto [70], Tesla Model 3 [74], and Range Rover Velar [75]. CarPlay and Android Auto mirror content from a mobile phone onto a car's inbuilt infotainment system. The Model 3 & the Velar have large touchscreens that are used for changing vehicle parameters and serve all multimedia needs. 24
- Figure 2.3: Google Lens (left) [72] is an Android application that allows a user to point his/her phone's camera at an object to get relevant information, Apple's Augmented Virtual

Display patent (right) [36] explores how VR content can adapt to the space a car is driving through.25

Figure 2.4: The Volkswagen Budd-E concept car has a large display on a passenger side window that is used for productivity tasks and video conferencing during travel.27

Figure 2.5: BMW’s AirTouch [68] and HoloActive Touch [47] concepts are designed to work with in-air gestural input from drivers and front passengers.....28

Figure 2.6: The SideTrigger gesture that was inspired by GSpeak’s trigger gesture [32], and shown to be a performant technique [1].30

Figure 2.7: Apple Maps’ [74] satellite view uses techniques first developed by Gleicher and Witkins [8] and refined later by Reisman et al. [35]. Using the widgets on the bottom right, one can peek behind a building (left, with red pin) occluding the PoI (right, marked with a purple pin).32

Figure 2.8. A rough estimation of the amount of time people routinely spend in informal (non-school) learning environments during their lifetime [5].37

Figure 3.1: A passenger door from a 2005 VW Passat mounted on a wooden frame. The assembly was designed to be easily wheeled around; and had height-adjustable wheels (this was also used to level the platform).....41

Figure 3.2: Switchable film that can be made transparent or opaque by passing electricity through it. This particular film was purchased from Smart-Tint [51].42

Figure 3.3: (From left to right) I removed the built-in window; attached a 0.1” thick acrylic sheet to the infra-red multi-touch frame to the window and stuck the switchable film onto this sheet; rear-projection on the switchable film while it is opaque; 65” TV screen

behind the door to show the external world (note, the height of the TV relative to the door was adjusted later).....43

Figure 3.4: The door window assembly with multi-touch input, and the option to toggle between opaque and transparent states because of the switchable film.....44

Figure 3.5: A solid state relay (SSR) and related electronic components (top-left). Arduino (top-right, inside a 3D printed case); the wiring was as follows: black to ground, the red wire to the Arduino’s 5V output, and the green wire to the Arduino’s digital input. A wire in the extension cord (bottom-right) was spliced and connected to the output of the SSR.....45

Figure 3.6: The Leap Motion controller [48] was mounted on the arm rest, but that was too low. It was moved closer to the window sill for more accurate tracking.46

Figure 3.7: Visualization from the Leap Motion Controller with the SideTrigger [1] gesture. ...47

Figure 3.8: Software architecture for AutoPoint’s lab setup. This architecture remained almost unchanged throughout the in-lab testing sessions, with changes limited to the visualization/UI layer only.....48

Figure 3.9: Bird's-eye view of the 3D world created in Unity (left), showing the path that the main in-game camera follows; the camera’s view of world (in-car perspective) at the Start point of the path/loop (right).....52

Figure 3.10: Interaction phases to select a target/PoI.54

Figure 3.11: A vector from the wrist to the top of the index finger was used to set the direction of a ray.....55

- Figure 4.1: Acrylic layer replaced with glass; it needed a sturdier aluminium mount (top-left, right), the switchable film now spanned the entire window (right), and the projector was mounted on a tripod by repurposing a plastic board.....63
- Figure 4.2: Tent poles connected to studio tripods (left) formed a structure that was used to drape a cloth over the car door to make a canopy (right). The window display was significantly larger than the prior iteration.....64
- Figure 4.3: A 4-point calibration step mapped the range of IR touch-frame coordinates to the projected display bounds.65
- Figure 4.4: A depth-of-field effect was implemented by blurring out everything in the scene other than the target (bluish grey building) closest to the camera (left). An area of focus (a bounding box) could be moved by a participant to jump to the next PoI (the mountain peak) by performing a pan gesture on the window.....67
- Figure 4.5: World Tilt: moving two fingers vertically on the screen gives the participant a bird's eye view of the scene.68
- Figure 4.6: Time Slicing: Take a snapshot of the external world by tapping the window (left). Then swipe anywhere on the screen to go backwards in time (right). In this case, the mountain had passed by already (1), but Time Slicing allowed it to be brought back into the frame (4).....69
- Figure 4.7: The target buildings now had a texture with a green highlight. I added smaller targets (red arrow) that could be easily occluded by other elements in the scene.....70
- Figure 4.8: The various combinations tested in V1 with each of the seven participants.....71

Figure 4.9: V1: Ranks assigned in order of preference to interaction techniques used to refine an Area of Interest. Lower scores are better (rank = 1 is the best, rank = 5 is the poorest).
73

Figure 4.10: The various combinations (per participant) tested in V2 with 24 participants.....76

Figure 4.11: V2: Ranks assigned in order of preference to interaction techniques used to refine an Area of Interest. Lower scores are better (rank 1 is the best, while rank =3 is the poorest). Number of rank = 1 for each technique are: Time Slice (13), World Tilt (8), Pan + Zoom (3).78

Figure 5.1: Initial design of the mount was an elongated s-shaped hook that supported the iPad on one end and hooked onto the top of the window on the other. Velcro strips on the back of the iPad kept it flush with the window.85

Figure 5.2: Blacked out rear window (top-left), a platform with a lip that the iPad rested against (top-right), and the iPad's camera aligned with a hole in the blacked out window (bottom).....86

Figure 5.3: iPhone used to video record the test session and provide cellular data to the iPad. ...87

Figure 5.4: Two distinct views, Live View (left), and Map View (right). Live View allows Time Slicing, taking a snapshot, and reviewing saved snapshots. Map View allows one to look at PoIs, explore by using pan and world-tilt gestures, and also review saved snaps.88

Figure 5.5: Time Slice initially shown as a small preview after pressing the "clock" icon. Panning/Swiping with one finger allows one to go back and forth in Time Slices. Tapping the preview makes it bigger. There also is a progress bar showing current position of Time Slices.....89

Figure 5.6: Map View showed a satellite view initially centered on the current location. A two-fingered pan/swipe (same as World Tilt in the lab) showed the 3D representation. PoI's were annotated with "A" markers. A user could save the map view or an image of the PoI.	90
Figure 5.7: The loop driven around and north of campus.....	91
Figure 5.8: Many participants devised a way to stabilize the hand before hitting a target.....	93
Figure 6.1: An overview of the various phases and outcomes from AutoPoint's development...	101

CHAPTER 1

Introduction

The experience of a road trip was perhaps first documented in the diary and associated photographs of Joseph Washburn during his family's travel from St. Paul, Minnesota to San Francisco in 1932. That diary spoke of the experience of high speed — for that era — travel along established routes. The scenic curiosities of mountains, canyons, and deserts epitomized Washburn's experience of the West as a region. In an analysis of Washburn's diary that was published in 1981, John Jakle [14] posits that Washburn's reaction can be attributed to the scenery that was curated for automobile accessibility. That is, the roads, the vehicle, and the drive itself shaped Washburn's image of the West in a way that was at times dramatically disconnected from the people, cultures, and ecosystems that Washburn drove past. That image of the West, as a distinctive and vast place, rich in natural wonders, was perpetuated by the popular media at the time, and that image has endured to this day.

1.1 Car travel colors a tourist's impression of a region.

The Washburns were city people and Joe, especially, was somewhat of a romantic in search of the excitement which life in a Midwestern city could not provide. Although he experienced the West as a tourist, he came away excited and felt compelled to author a travel diary. The manuscript, entitled "Western Trip July 31-September 3, 1932," comprises twenty-eight pages of typewritten script and eighty-eight photographs. The road trip seemed to have served Joe's needs for an adventure, an adventure enhanced by the challenge of recording his experiences through the then demanding medium of photography.

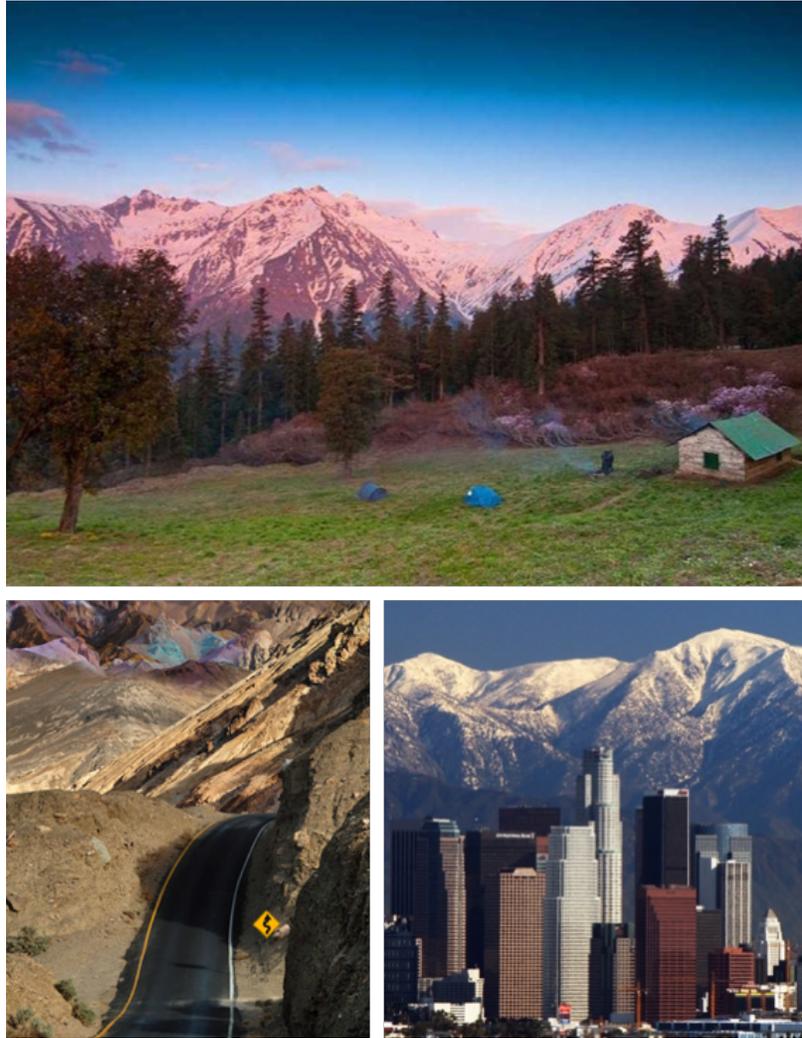


Figure 1.1: The Himalayas in India seen from a country road (top), winding roads through Death Valley with geological landmarks (bottom left), and the LA skyline with the San Gabriel mountains in the background (bottom right); all sights and features one can encounter on a road trip.

Many decades later, my first impression of the Indian Himalaya was formed during a road trip I took with my family in the late 1980s. My father, an amateur mountaineer, pointed out mountains that we saw along the way, identifying them by name and regaling me with tales of how some of them were climbed. The spectacular sights and narratives on that road trip (and others thereafter) left an impression on me. Though separated by geography, a few decades, and a travel diary, Washburn's experiences were remarkably aligned with mine.

Such experiences are common. Road trips are popular, and they have enriched travel experiences. Other modes of travel take us to an end destination; a car trip can be an end in itself. That being so, I thought of ways of making it more engaging, more interesting, of ways to forge a richer connection with the spaces we pass through. Each road trip comes with its own set of landscapes (Figure 1.1). We record the sights and sounds of such journeys in photographs and videos and then go home. Instead of a still photo or a video recorded while traveling through interesting places, wouldn't it be far more interesting to engage in real time with what we see out of a car window? Can the automobile itself scaffold those interactions? What may such a system look like? Those are the questions I address in this dissertation.

1.2 Current and future Automobile systems

For many people, cars are more than individual means of transport; they value their cars as personal spaces [18]. In countries all over the world people spend significant time in their cars while commuting to work, running errands, or driving for pleasure. Vehicles have become a hub for information access, communications, media consumption and personal entertainment. Even consumer mobile technology has spilled over to automobiles in the form of Android Auto [70] and Apple CarPlay [71], which mirror the contents of mobile phones onto car infotainment systems. Most of these systems are highly driver oriented and are designed to limit distractions while driving. Naturally, these design constraints limit the duration and richness of interactions. However, with self-driving cars just around the corner [46], there exists a great potential for in-car interface design to move beyond driving and navigation assistance focused solely on the driver.

Despite that potential, most automobile manufacturers have been focusing on taking select parts from our daily experiences and shoehorning them into an automobile. For example, Ford's

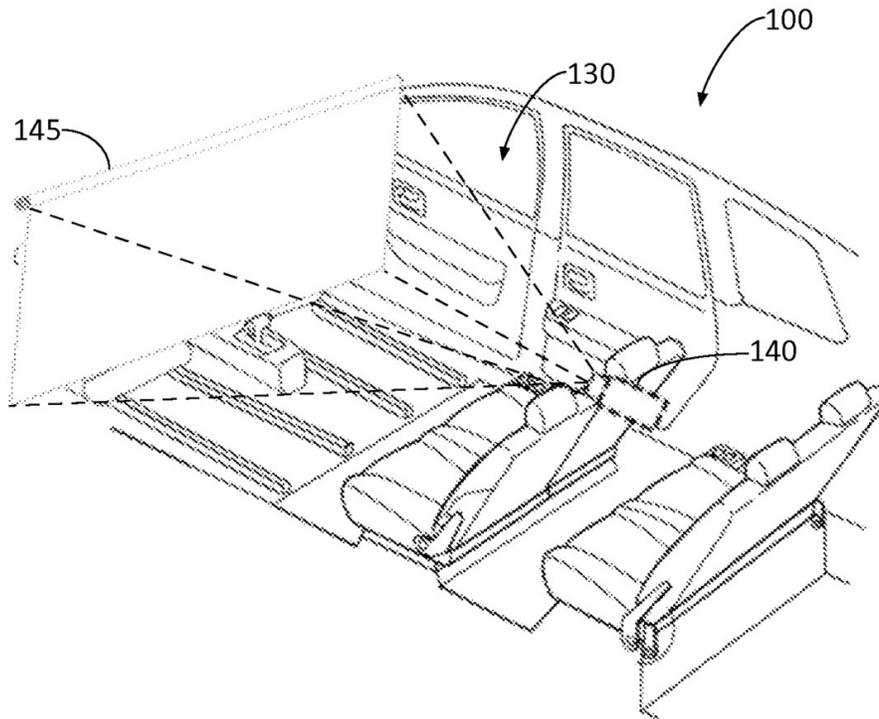


Figure 1.2: Ford has outlined its vision of in-car entertainment for self-driving cars where a large projector screen that displays multimedia content is positioned in front of the passengers [6].

vision of in-car interface design involves taking the living room and movie theatre experience to the car (Figure 1.2).

This is where we encounter competing visions - one vision says that cars become entertainment, that it is all about killing time and getting to the destination. On the other hand, my intent is to explore ways to connect passengers with the space they are traveling through.

One can argue that car travel for most people is confined to the unremarkable urban sprawl, and that audio-visual entertainment relieves boredom and is the optimal solution. A counter argument would be that such designs shortchange the experience of traveling on roads that have views that can captivate us. I claim that the automobile in its modern guise should be able to straddle both travel in the urban sprawl and enhancement of the experience of a road trip worthy of a guided tour.

With the latter scenario in mind, I have developed AutoPoint, a combination of technologies and interaction techniques designed to enable passengers in a car to engage with Points-of-Interest (PoIs) outside their vehicle (Figure 1.3). These include buildings, natural formations and ecosystems.



Figure 1.3: A concept sketch of AutoPoint, where a user interacts with the nearest car window using multi-touch input or in-air gestures detected via a Leap Motion Sensor [48].

With AutoPoint, a car's window serves as an interactive and transparent display surface. A passenger gives input to the system via multi-touch, and in-air gestures. Broadly speaking, I divide AutoPoint's interactions into two distinct phases: a) selection of a general area of interest, and b) a refinement of that selection to a particular PoI.

The general area of interest is selected by taking a snapshot – from the point of view of the passenger – of the space the car is travelling through. This snapshot is taken by tapping or pointing

at the car window. Since the number of potential PoIs within a snapshot varies with the scene being driven through, the speed of the car, and the proximity of PoIs to the car, homing in on the target requires additional inputs from the passenger.

During the subsequent refinement phase, the passenger can narrow down that area of interest/snapshot into a particular PoI using interaction techniques that involve multi-touch input on the car's side window. Once a PoI is selected, additional contextual information— e.g. data about a building — is provided on the car window as well.

AutoPoint went through three design iterations in the lab. With the first iteration, I made a proof-of-concept prototype by temporarily mounting a car door on a portable platform and used a projector to project images on its window (Figure 3.4). This prototype was tested with five participants to gauge their initial reactions, and to iron out any glaring design pitfalls.

For the second iteration – tested with seven participants – the hardware setup was refined significantly (Figure 4.2), and several multi-touch interaction techniques intended to be used for AutoPoint's PoI refinement phase, were added to the system. In its third and final iteration in the lab, the software underwent a redesign, and the better performing interaction techniques shortlisted from the second iteration were selected for an in-depth evaluation. This final version was tested in the lab with twenty-four participants.

Following the lab study, I developed a version of AutoPoint as an iPad application. This application incorporated the features that were shown to work well or preferred by the participants in the lab. I fitted a car with an iPad that mirrored the lab setup as closely as possible (Figure 5.2) and drove 14 participants on a loop around campus to test the system in the real world and get participant feedback. In aggregate, AutoPoint in its lab and real-world variants was tested with fifty participants.

In this dissertation, I explore the design space of a class of problems I like to refer to as Multimodal Transparent Display Interactions (MTDI). Cars with windows as transparent displays add another dimension to MTDI, one where the external world as seen through the display is in motion. With AutoPoint, I explore a problem domain that encompasses MTDI-EWM (MTDI with External World in Motion). This problem domain has various characteristics: it is temporal, geo-spatial, location based, and is also social in nature.

I investigate multi-touch and freehand point-and-select interactions previously developed by the HCI research community, and their application in Automotive UI design (as a subset of MTDI-EWM). I report on the design, implementation and evaluation of AutoPoint. I also claim that interactive car windows are a feasible way to engage with the world one is driving through. While this engagement with the outside world could be purely for entertainment, AutoPoint seems to hold promise as an informal learning environment; to quote one of my participants: “this is like a mobile museum”.

CHAPTER 2

Related Work

The modern automobile is a complex mechatronic system [30]. Computers control all sorts of automotive functions ranging from vehicle performance to in-car entertainment and connectivity. This prevalence of computers has also altered car interiors. Electro-mechanical systems with switches and dials have given way to a display-rich environment. It is not far-fetched to think that car windows would evolve to become transparent displays. AutoPoint's design space falls under a class of problems I like to refer to as Multimodal Transparent Display Interactions (MTDI). Cars with transparent displays add another dimension to MTDI, one where the external world, as seen through the display, is in motion.

With AutoPoint, I explore a problem domain that encompasses MTDI-EWM (MTDI with External World in Motion). This problem domain has the following characteristics: a) Temporal: with a vehicle in motion, there is a timing component in trying to interact with something a user sees through a transparent display; b) Geo-Spatial: the external world is complex and it is non-trivial to figure out which object a user intends to target; it could be very close, miles away, or somewhere in between along the line of pointing; c) Location or Map based: the density of potential targets in the external world can vary between locations; d) Social Nature: automobiles can have co-located users, and a transparent display could be a shared resource. I also explore whether applications built with those characteristics in mind can help transform a car ride into an informal learning environment. With that in mind, AutoPoint seeks to contribute to the following areas of research: (1) automotive user interfaces; (2) content navigation using multi-touch and in-air input modalities; (3) in-car interaction design; (4) cars as informal learning environments.

2.1 Automotive User Interfaces

The design of past, current and future in-car user interfaces is a clear area of interest. In this section, I first take a brief look at car dashboards from the past and their evolution up to the late 80s. Thereafter, I report on the current state of in-car infotainment systems and prototypes and industrial concepts (usually on display at auto shows) that offer an early peek at the direction in which car manufacturers are heading.

2.1.1 *Car dashboards and interfaces from the past*

The Ford Model T was voted the most influential car of the 20th century [64,66] and is also regarded as the car that helped middle-class America travel. Looking at the Model T's dashboard (Figure 2.1, top-left), we can see that the only dial (or measurement device) is an ammeter, and the rest are mostly mechanical linkages needed to drive the car.

The Tucker Torpedo from 1948 (Figure 2.1, top-right), a remarkably forward-looking design [65] manufactured in Chicago, had a much more advanced human-machine interface compared to the Model T. There were many electro-mechanical switches and electro-hydraulic systems with interfaces controlled by the driver. Particularly worth mentioning is the switch for the “Cyclops Eye” lights that turned when the car's steering wheel was turned greater than 10-degrees (to help with visibility around turns), and the 4-speed column mounted manual transmission that was hydraulically assisted.

We see a shift in the design from the 1950s onwards. Cars had radios as optional equipment. In fact, air conditioning, cup holders, and radios on the dash emerged (Figure 2.1, bottom right) as cars were being used to commute to work, and for longer road trips. Drivers demanded comfort and entertainment options over and above utilitarian features.

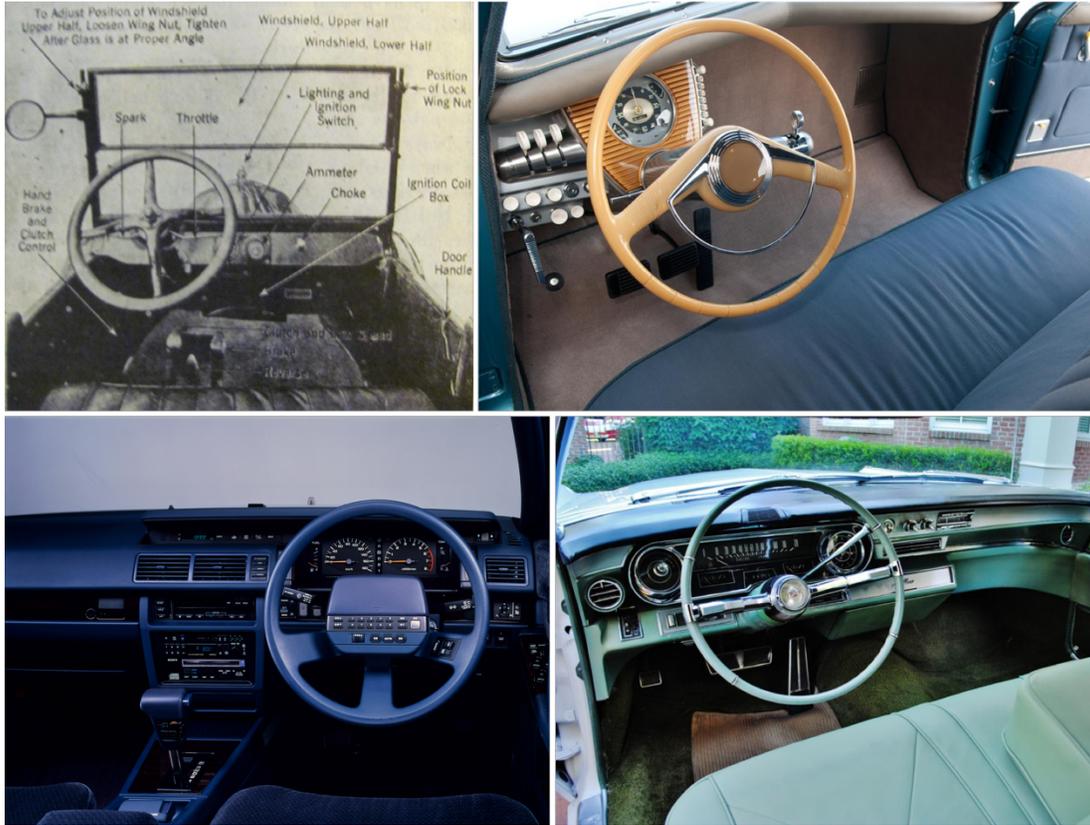


Figure 2.1 (clockwise from top-left): 1923 Ford Model T [64,66] with an ammeter as the only dial on the dashboard; the 1948 Tucker Torpedo [65] with a lot more dials and switches; the 1965 Cadillac Deville [63] with an A/C and radio on the dashboard; the 1988 Nissan Heritage [33] with a car phone and digital displays driven by an onboard computer.

In the mid-80s, as technology evolved, digital displays and phones started making their way into cars. For example, the 1988 Nissan Heritage (Figure 2.1, bottom-left) had a dashboard mounted digital display that showed the climate control information, the time, the average fuel efficiency etc., and it also had a car phone mounted on the center console. The phone's dial pad was on the center hub of the steering wheel (this center hub was fixed, while the rest of the wheel turned freely behind it).

It almost goes without saying, but the evolution of car interiors coincides with major political and technological changes, from fighter jet like interiors after WW-II, to cars with A/Cs and radios, and finally to cars with digital displays powered by on board computers.



Figure 2.2: (clockwise from top-left) Apple CarPlay [71], Android Auto [70], Tesla Model 3 [74], and Range Rover Velar [75]. CarPlay and Android Auto mirror content from a mobile phone onto a car's inbuilt infotainment system. The Model 3 & the Velar have large touchscreens that are used for changing vehicle parameters and serve all multimedia needs.

2.1.2 Current in-car infotainment systems and display technology

Since the early 2000s, in-car displays have moved from simply revealing information visually, to becoming actual human-machine interfaces. This shift, along with the increased volume of displayed data, is driving the need for displays of larger sizes, non-rectangular or curved shapes, as well as higher resolutions.

With Apple CarPlay [71] and Android Auto [70], content and applications on your mobile phone are mirrored on the car's built-in display (Figure 2.2, top-row). Maps, making phone calls, and entertainment in the form of music/podcasts/radio are the most commonly used applications [18]. Input modalities with CarPlay and Android Auto include voice assistants (in the form of Siri,

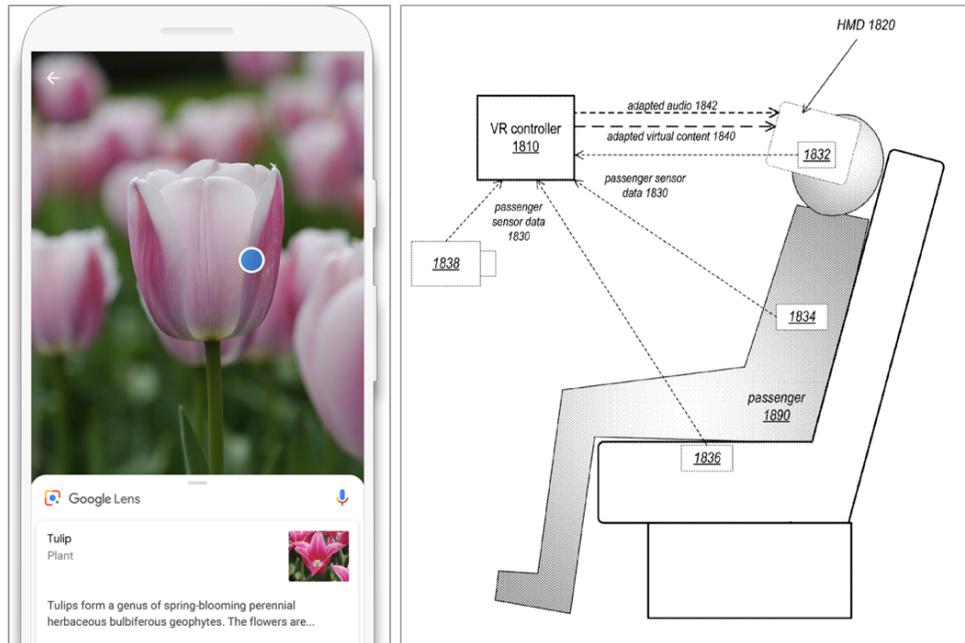


Figure 2.3: Google Lens (left) [72] is an Android application that allows a user to point his/her phone's camera at an object to get relevant information, Apple's Augmented Virtual Display patent (right) [36] explores how VR content can adapt to the space a car is driving through.

and Google Assistant respectively), and multi-touch input. CarPlay also supports input from all the knobs, dials, or buttons in the car, including the ones on the steering wheel.

Some cars, like the Tesla Model 3, have replaced almost all physical controls with a touchscreen (Figure 2.2, bottom-right); a 15-inch touchscreen display controls all vehicle parameters and infotainment needs. On the other hand, the Range Rover Velar's dashboard has preserved a more conventional appearance, but it has large dual touchscreens (both 10-inches measured diagonally) with embedded physical controls (Figure 2.2, bottom-left). This split design allows individuals to view and interact with multiple features simultaneously, e.g. use the navigation system in the upper screen while playing multimedia content on the lower screen.

Most of these modern infotainment systems are driver focused; rarely do cars have options designed primarily for passengers [26,27]. As cars become more autonomous, it seems likely that

these designs would evolve to cater to all passengers equally. In addition, driver aids in modern cars, such as a lane departure warning system, radar cruise control, an automatic collision avoidance system, and even the increasingly common reversing assist system, require cars to be fitted with sophisticated sensors and cameras. With this sensor network that continuously scans the outside world, and plenty of computing power on board, we already have the building blocks to make a system that can connect passengers in a car to the space they are traveling through.

Apart from touchscreens, other display technologies adapted for cars include Head-up Displays (HUDs) that have commonly been used to provide contextual information to drivers and passengers. General Motors was the first to use a single color laser projector to mark up traffic signs on the windshield [7]. In the research community, interactions with HUDs were examined by Milicic et al. [28], who conducted a study in which subjects were able to complete menu interaction tasks faster when using a HUD in comparison to a head down display. Lauber et al. used a HUD to mirror both the input and output of a touch screen located in the car's center stack [19]. BMW claims that HoloActive Touch [47] (Figure 2.5, right) is more advanced than existing heads-up displays, and it uses reflections to create the illusion of a display floating in space rather than on the windshield.

Similarly, Augmented Reality (AR), and Virtual Reality (VR) based content and visualization techniques that are being explored by technology firms [36] (Figure 2.3), can also be adapted for cars. Apple was recently awarded a patent [72] for a system that is designed to provide immersive VR experiences for passengers by mapping a vehicle's motion to the virtual world. As an aside, the patent application mentions something similar to what I put forth earlier in this document: "a VR system with a vehicle in motion provides opportunities for enhanced virtual experiences that are not available while sitting in a room using a stationary simulator or any head-mounted device".



Figure 2.4: The Volkswagen Budd-E concept car has a large display on a passenger side window that is used for productivity tasks and video conferencing during travel.

That said, between AR and VR, AutoPoint’s design and implementation leans towards AR, with the passenger being completely un-instrumented (i.e. no VR goggles to wear, nor having to use specialized input controllers), and making use of a car’s window as a transparent display.

2.1.3 Industrial Design Concepts

The Volkswagen Budd-e [67] (Figure 2.4) was on display at CES 2016. According to Volkswagen, the Budd-e is a mobile interface that connects the vehicle with the world around it. It bills the car as an integral automotive component in the Internet of Things (IoT); it is capable of communicating with a Smart Home or office. The Budd-e employs touch and gesture controls for user input instead of switches and buttons inside the cabin. It has individual displays that blend



Figure 2.5: BMW's AirTouch [68] and HoloActive Touch [47] concepts are designed to work with in-air gestural input from drivers and front passengers.

into large infotainment panels, and digital screens instead of analog mirrors and windows. Of these displays, the one on the rear passenger window is of note. It is a large high-resolution display (Figure 2.4, inset). In this version of the concept, Volkswagen showed this display being used for productivity tasks and video conferencing during travel. With AutoPoint, I focus on a passenger side window as an interactive surface, and this is one of the first concepts that seem to be based on that vision.

BMW's AirTouch [68] (Figure 2.5) allows the display in a vehicle to be operated like a touchscreen without actually having to make contact with a surface. High fidelity motion sensors are installed on top of the dashboard and these sensors respond to hand movements to offer three-dimensional control. While these in-air gestures can activate and help navigate through menus on the display, AirTouch has a button located on the rim of the steering wheel that can be pressed to confirm the initial selection gesture (e.g. selecting a particular list item after scrolling through it using in-air gestures). The front passenger also has a similar button positioned on the side sill in the

door pad. Passengers can also use one hand to navigate through the menu via in-air gestures, and the other hand to confirm inputs.

While this system was designed for the driver and the front seat passenger, scenarios where passengers riding in the back seat could use multi-modal input techniques seemed interesting to me. For example, if a passenger side window were to be interactive, could a passenger sitting in the middle, or on the other side, interact with that particular window via in-air gestures instead of trying to reach across other people?

2.2 In-air gestures, and 3D content navigation via a 2D input device

In this section, I cite research that informed the design of AutoPoint’s input techniques. Direct, multi-touch devices have been shown to be effective input modalities [21,41] that are consistent with our “natural behavior” [29]. There is a large volume of research on multi-touch interaction with computer displays that spans more than three decades [21]. Relatively recently, multi-touch and in-air gestural input has been classified under natural User Interfaces [29,41]. In-air interactions have also been researched for over twenty-five years [13], especially in the context of VR and large displays [31] where touch input can be impractical due to display size and distance from the user.

While the size of a car’s window should not rule out touch input, a passenger’s position in a car is typically constrained by a seatbelt. If a passenger wishes to interact with a window – e.g. because of a PoI seen through it – there is a chance that it might be out of reach. In such a scenario, in-air gestures can be convenient. Aside from in-air gestures, using the car’s window for multi-touch input seems logical. Since the window is a 2D surface, and the world outside is 3D, interaction techniques that were designed for similar scenarios are relevant and discussed briefly.

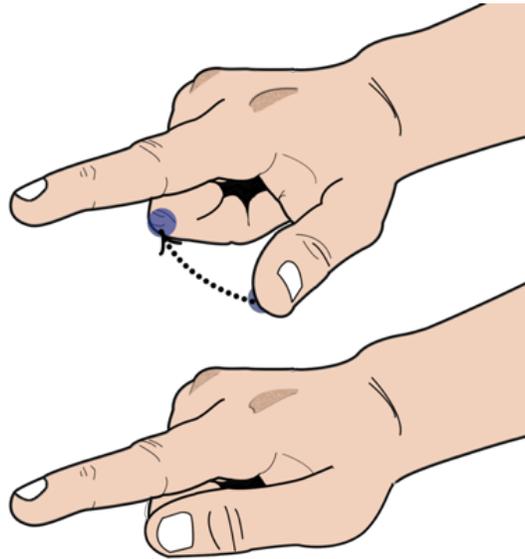


Figure 2.6: The SideTrigger gesture that was inspired by GSpeak’s trigger gesture [32], and shown to be a performant technique [1].

2.2.1 Free-hand pointing and in-air gestures

Device-less interaction techniques usually rely on computer vision to detect hand and finger movements, either through markers placed on the hands, or by relying on sophisticated machine learning models for hand movement [48]. A major advantage of such vision-based techniques is their ability to track multiple fingers uniquely [4,40]. However, these in-air interaction techniques lack explicit inputs, such as buttons, making selection techniques and clicks non-trivial. Wilson [43] used pinching as a technique for cursor control through robust marker-less computer vision. However, interaction was limited, and required the gesture to be performed over a pre-set background (a keyboard), with a close-range camera.

As an alternative, the Head Crusher technique casts a ray from the user’s eye through the point midway between the user’s forefinger and thumb, and onto the scene [34]. The object is acquired when it intersects with the ray. Vogel & Balakrishnan [40] explored single hand pointing and

clicking interactions with large displays from a distance. They found ray casting to be an effective pointing method, and proposed AirTap as a clicking gesture for single clicks.

Remote bimanual multipoint systems include the g-speak spatial operating environment [32] and virtual reality applications using Pinch Gloves [73]. In g-speak [32], the user points at a target by making a trigger gesture (finger pointed towards display, thumb towards the ceiling), and selects by lowering the thumb on top of the index finger [45]. The gesture that I used for AutoPoint's area selection phase is the SideTrigger gesture [1] (Figure 2.6) whose design was informed by these techniques.

Variations of some of these in-air interaction techniques were also explored by Toppan and Chiesa [39]. They used a Leap Motion sensor to add in-air interaction capabilities to the infotainment system in a car. The authors implemented several gestures, for example, swipe left or right to navigate among menu pages (similar to BMW's AirTouch [68]); key tap (similar to Air Tap [40]) to perform a selection; an anti-clockwise rotation of the hand to go to a previous page in the menu; and went as far as writing letters in the air with a finger to establish a destination on the GPS.

2.3 Navigating 3D Spaces using a 2D input device.

I imagine that users of AutoPoint's interaction techniques would be able to interact with 3D models of buildings and other objects in the landscape that they're traveling through. Since these interaction techniques involve the car's window – a 2D surface – I briefly review some relevant research in this area.

Mackinlay et al. developed the Point Of Interest desktop technique [24] that was designed for fast and controlled movement through a 3D space using a 2D input device (a mouse at the time).

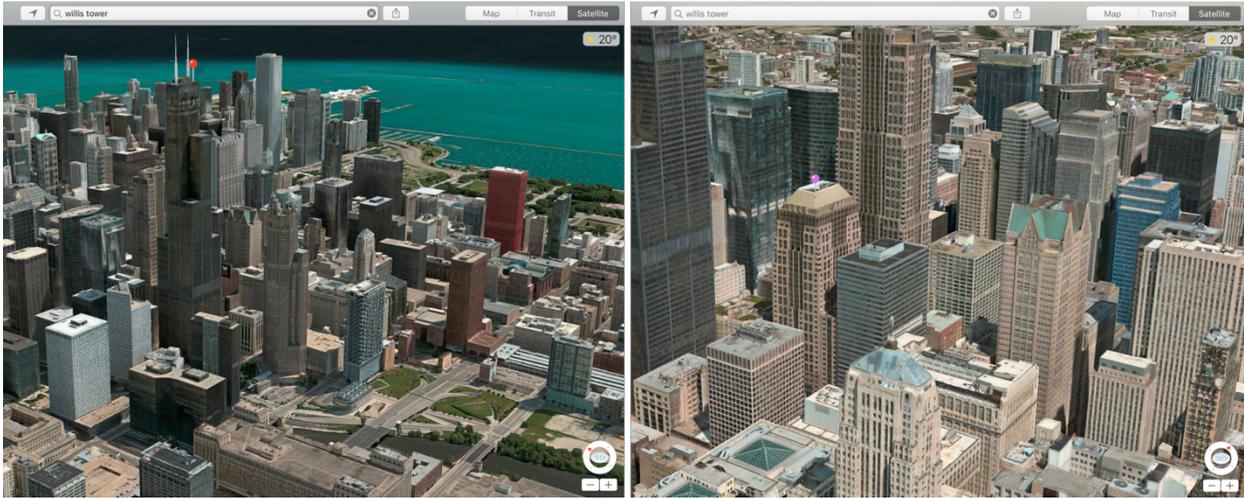


Figure 2.7: Apple Maps’ [74] satellite view uses techniques first developed by Gleicher and Witkins [8] and refined later by Reisman et al. [35]. Using the widgets on the bottom right, one can peek behind a building (left, with red pin) occluding the PoI (right, marked with a purple pin).

A user indicates a point of interest (target) on a 3D object and the technique uses the distance to this target to move the viewpoint logarithmically, by moving the same relative percentage of distance to the target in every animation cycle. The result was rapid motion over distances, slowing as the viewpoint approaches the target object.

With Unicam [44], Zeleznik et al. devised a method of using a virtual 3D camera with a 2D mouse or stylus input device that was based on gestural interaction. UniCam required only a single-button stylus or mouse to directly invoke specific camera operations within a single 3D view. It supported translation in three-dimensions, orbiting about a point, animated focusing on an object surface, animated navigation around an object, zooming in upon a region, etc.

Navidget [10] by Hachet et al. built upon the PoI and Unicam approaches by providing better feedback and control for camera positioning and it was tested and found to be especially useful for inspecting targets at a distance.

Unsurprisingly, if a target is occluded, it can only be manipulated indirectly (e.g. you cannot click or touch a building hidden behind another building). Given the ability to manipulate the

whole scene, a user can reveal the PoI. An approach for this was first proposed in Gleicher and Witkin's early work on through-the-lens camera control [8] and was more recently adapted by Reisman et al. to manipulate 3D content on 2D multi-touch systems [35]. Apple Maps has used this technique for navigating a scene in its satellite view mode (Figure 2.7).

2.4 In-Car games and interaction design

In-car interactions have been receiving growing attention from the HCI community [3,17,18,23,27]. Meschtscherjakov et al. [26] divided a car's interior space into three design areas: the driver's area, the front seat area, and the backseat area. They noted that most in-car systems have been focused on the driver's area, and passengers occupying the front and back seat areas have very different requirements than those of the driver. I think it is important to acknowledge that self-driving cars *could* signal a change in the seating configuration of a car (e.g. passengers sit facing each other, instead of the current front-to-back seating); for now, however, I assume that the current arrangement remains popular since most people prefer to be seated facing the direction of travel.

Laurier et al. [20] noted that people sometimes delight in traveling in a car together. For families in particular, cars have become significant places for parents to learn about and dialogue with their children, and for children to learn from their parents. Since, the car is a setting where passengers are assembled closely together, many conversations that were reserved for dinner tables have also seen a shift into the space of the car.

Wilfinger et al. [42] conducted a cultural probing study investigating the activities and the technology usage in the back seat of cars. They found that fighting boredom was one of the most important issues for all passengers. Mobile phones, in particular, were considered standard

equipment in the back seat, mainly for older children. The authors concluded that “integrating aspects of education into an in-car game is a promising approach, especially when it is aimed at topics related to the current trip”.

There have been several in-car games that have been developed in recent years. “Backseat Playground” [2] is a game that presents sequential audio stories based on a player’s location. As the car travels through a landscape, a crime story unfolds. A player takes part in the story by using a set of tools on a hand-held device. Participants in a user study indicated that they felt like the events described as a part of the game’s narrative genuinely took place outside the car.

Wilfinger et al. [38,42] also developed a concept for a back seat entertainment system called “RiddleRide” — a multiplayer context-aware quiz application aimed to entertain and educate passengers during rides. RiddleRide poses questions via speech, for example: “what do you see on the hill to your left?”. Potential answers were adapted to different age or skill groups and they were displayed on mobile devices. After each player selected an answer, the system revealed the correct answers and the current score. Sundström et al. [38] presented three in-car games aimed at making sitting safely more enjoyable for children. The goal of the first game, called “RainbowBalance”, is to balance a virtual ball using head movements. Children can see the results of their movements on a screen attached to the front seat. In the second game, called “emoCar”, the driving style of the actual car determines the direction of a car avatar, driving on the roads between ‘happiness’, ‘anger’ and ‘sadness’. Children need to catch the avatar by performing the appropriate facial expression for the desired direction. The third game, called “GhostCatcher”, consists of an actual jar that children hold in their hands. Their goal is to open the jar only while the car is exposed to darkness (e.g., while driving through a tunnel). The jar then starts to vibrate and emit sounds, signaling that ghosts have been captured. If the jar remains open when the car is exposed to light,

the jar goes silent and still, signaling that all ghosts have escaped. A user study with children revealed that GhostCatcher was a very enjoyable game, and quite well suited for connecting children to the car and the outside world.

Hiah et al. [12] presented a concept of a robot companion that was designed to address the needs of young children sitting in the back seat. It was placed between the two front-seats, and essentially served as a virtual guide that allowed children to tour through various sights and learn fun facts about them. A preliminary user study employing a “Wizard-of-Oz” methodology showed that children seemed to be engaged when interacting with the companion robot. “nICE” [3] is a multimodal, collaborative game, played on two multi-touch tablet devices by the passengers in the front and back seat. During the game all occupants of the car are tasked with guessing the contents of hidden images based on several image snippets and audio hints. These snippets and hints can be unveiled by playing different collaborative mini-games. A follow-up study indicated that users were highly focused on the game, neglecting the outside world.

While games are clearly of interest to me as an application area for the AutoPoint system, there are a couple of research projects that are also aligned with AutoPoint’s research goals. Rümelin et al. [37] investigated pointing as a form of gestural interaction in cars. In a pre-study, they showed the technical feasibility of reliable pointing detection with a Kinect depth camera by achieving a recognition rate of 96% in the lab; note that this number represents a successful pointing gesture recognition, and not target selection accuracy. They did a subsequent real-world study that was more observational in nature, without a sensor detecting gestural input. Participants were asked to point to objects inside and outside of the car while driving through a city. The researchers observed how pointing changed driving behavior, e.g. whether drivers slowed down more than usual. They used the car’s on-board diagnostics (OBD-II) port and an eye-tracker to measure driver distraction.

Matsumura et al. [25] proposed a method to stick in-car conversation into places (think of it as audio graffiti) by recognizing finger pointing gestures from multiple passengers in a car. They used a motion capture device to detect finger-pointing gestures and tried to identify a PoI by calculating the intersection of these finger pointing gestures. Their system had a few limitations: firstly, it assumed that there were multiple people pointing at the same object, and secondly, there was no feedback or interaction technique that could refine a general area of interest into a particular PoI.

This discussion of pointing based in air-interactions would not be complete without a note about Goodwin's research on Pointing as a Situated Practice [9]. Goodwin draws out the subtle cues that are embodied in a pointing gesture. The gesture points towards a particular place -- which Goodwin calls the domain of scrutiny -- where an addressee would look to find the target of that gesture, i.e. the particular entity being pointed at. A single domain of scrutiny can contain multiple targets. Goodwin notes that there is an interlocking between the pointer and the addressee that consists of cues like body posture, body position, and speech. All of these cues when taken together, help narrow down the actual target of a pointing gesture.

I will come back to this later, but AutoPoint's in-air pointing gesture to select an area of interest (domain of scrutiny), followed by a narrowing down of a number of possible targets to a single target via multi-touch input on the car door, meshes well with the framework presented by Goodwin.

2.5 Cars as informal learning environments

Informal learning environments include a variety of settings ranging from family discussions at home, visits to museums, to everyday activities like gardening, as well as recreational activities like hunting or hiking [5]. Since we spend a large proportion of our time in informal spaces (Figure

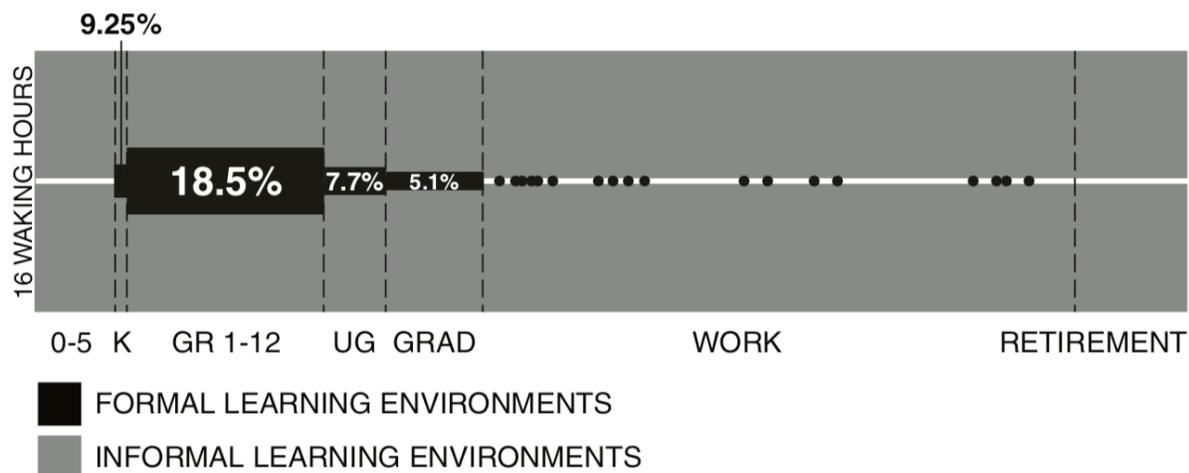


Figure 2.8. A rough estimation of the amount of time people routinely spend in informal (non-school) learning environments during their lifetime [5].

2.8), everyday experiences support learning for most people. A committee for the National Research Council traced the evolution of informal learning environments in the US over the last hundred-years. In this report [5], the authors note that over the years, technological advances have distanced people from traditional agrarian experiences. That even though the invention of communication devices like radios, the Internet, and more recently, smartphones, have increased the availability of information, modern living has distanced society from the natural world.

The NRC report also mentions that the "material world", with its rich location-specific features and processes, can become the focus of inquiry and learning. The authors note that this is already true of niche groups that study the interdependencies of natural systems (e.g. ecologists) and the influence of humankind on the environment. The authors define this kind of learning as learning through a "Place-Centered Lens". That is, learning that happens within and across locations.

Car rides, by their very nature, qualify as place-centered informal learning environments. The same way growing up on a farm tends to connect children to flora and fauna due to frequent exposure, perhaps allowing children to engage with the eco-systems they are being driven through

would help alleviate some of the disconnect with the natural world. How we can design a car-based system that could potentially serve such a function, is described in the next few chapters.

CHAPTER 3

AutoPoint Phase - I

The core idea of AutoPoint was to engage with the world one is driving through. Across the various phases of the project, I investigate the value and viability of ways to implement this idea, and report on the various HCI challenges and constraints faced along the way.

The starting idea was to represent the outside world, as seen through the car window, in a medium that would permit, first, the recognition of a broad area of interest in that world and then, the unequivocal identification and selection of a point of interest (PoI) within that area. Our eyes see a 3D world but for the car passenger, the representation in question was to be on a 2D plane, such as the car window itself.

The first phase consisted of building a laboratory setup to simulate a car-like environment. This setup would have to be capable of not only casting on the window real time snapshots of a moving world, but also of switching to a target selection and data presentation mode upon the selection of an area of interest. The laboratory setup would have to incorporate the hardware capable of achieving these goals and the software to drive it. Thus, the design of the setup consisted of two elements, the conceptualization and embodiment of the relevant hardware, and the architecture of the software to go with it.

3.1 Hardware Design and Assembly

The original concept for AutoPoint (Figure 1.3) was to allow for multi-touch and in-air gestural inputs on a car window that was also a transparent display. In this concept, a multi-touch enabled computer display was to be mounted directly on the window frame of a car door. Very quickly, it

became apparent that this concept had serious issues; the lack of any depth of field, or any room to allow for parallax (which is a given for a real car window) meant that it was not worth pursuing this strategy.

The ideal replacement would have been a transparent OLED display [69], but transparent displays are a couple of years away from being commercially available. In addition, the displays that were showcased as concepts at the most recent Consumer Electronics Show [49] lacked support for multi-touch input. I chose to go with the option of using a projector to cast content on the car door. The re-imagined representation of the world, for the laboratory setup, would be on a display (TV screen) outside the car's window. That representation, in turn, would have to be rendered on the car window, by projection, for the PoI selection stage. Further, the image on the car window would have to have interactive capabilities. At this stage, the overall design concept for the work flow was:

- User sees passing scene outside the car window (video running on the outside display/TV screen)
- User identifies an area of interest and selects it by pointing or by touch input on the window surface
- The projector immediately casts a snapshot of the outside scene onto the window.
- The freeze frame enables the extraction of data about the item of interest.

3.1.1 Car door mount, projection surface, and projector setup

I purchased a car door from a local auto-salvage yard. There were only a few of things that I had to consider while selecting one: a) it had to be a right-rear door because that was a better fit in the lab space where I was planning to set it up, and, b) by choice the window's design had to be fairly



Figure 3.1: A passenger door from a 2005 VW Passat mounted on a wooden frame. The assembly was designed to be easily wheeled around; and had height-adjustable wheels (this was also used to level the platform).

common to ensure that the size and shape of the window was not particularly unusual (i.e., not too narrow or too small/large).

To mount the car door, I built a wooden platform similar to a moving dolly (Figure 3.1). I cut a slot on one side of this platform for the bottom edge of the door to fit into. The door was sandwiched in place with supports on both sides. The support on the inner side (i.e. the side a passenger would sit) was adjustable to help keep the door secure and vertically aligned. Similarly, the caster wheels were mounted on bolts that permitted a couple of inches of height adjustability and allowed me to align the door relative to the projector and the display I now planned to use in the background (the TV screen).

For the interaction stage, the first attempt was to have the car's window mimic a transparent display. Since it had a light green tint to it, it had seemed possible that a clear image might form on it, without too much reflection. At other times, when viewing the video display on the TV screen outside the car, the plan was to project a fully black image onto the glass; the black image was



Figure 3.2: Switchable film that can be made transparent or opaque by passing electricity through it. This particular film was purchased from Smart-Tint [51].

intended to obliterate the projected visual. In the event, there was too much glare and reflection from the glass and this plan was abandoned.

I tried using various kinds of plastic films applied on the window to make the surface more suitable for projection, but it was difficult to find the balance between a film that was transparent enough and also diffused the light evenly. In addition, I also needed to add multi-touch capability to the system, which would have been harder on the slightly curved built-in window.

For multi-touch input, I had planned to use an infra-red (IR) touch frame [50] fitted around the car window. There are many companies that manufacture such frames, and they are commonly installed as an overlay on any ordinary display.

The revised design consisted of a clear acrylic sheet, 0.1” thick, affixed to the IR touch frame so that the assembly could replace the car window (Figure 3.4). Since this acrylic sheet was too reflective by itself, I looked at other materials that could be used as a projection surface. I found a switchable film (Smart-Tint [51]) that was perfect for my application; this film has liquid crystals (Polymer Dispersed Liquid Crystal (PDLC) [52]) enclosed between two layers of ITO [16]. When a current is applied to this film, the liquid crystals align and allow light to pass through. When the current is cut off, the liquid crystals are oriented randomly, and that makes the film opaque. While



Figure 3.3: (From left to right) I removed the built-in window; attached a 0.1" thick acrylic sheet to the infra-red multi-touch frame to the window and stuck the switchable film onto this sheet; rear-projection on the switchable film while it is opaque; 65" TV screen behind the door to show the external world (note, the height of the TV relative to the door was adjusted later).

it is opaque, the film diffuses light in a way that makes it a satisfactory projection surface (Figure 3.2 and Figure 3.3).

With the projection surface in place, the next question was about how to carry out the projection. The role of the projector was to present a snapshot of the most recent scene from the video of the outside world. The only feasible option was a rear-projection setup (i.e. the projector is placed behind the projection screen) because a ceiling or top-mounted projector would either shine a bright light on the user or occlude the image on the window frame while a user was trying to interact with the system. I chose a BenQ [53] short-throw projector with a high-degree of lens-shift and keystone control because it had to be placed quite close to the car door and project at a steep angle (Figure 3.3, middle-bottom).

From a user's point of view, the IR touch-frame with the acrylic sheet was a satisfactory substitute for the built-in window. The projected image was clear and sharp (Figure 3.3, right) without any risk of the light from the projector blinding a user.

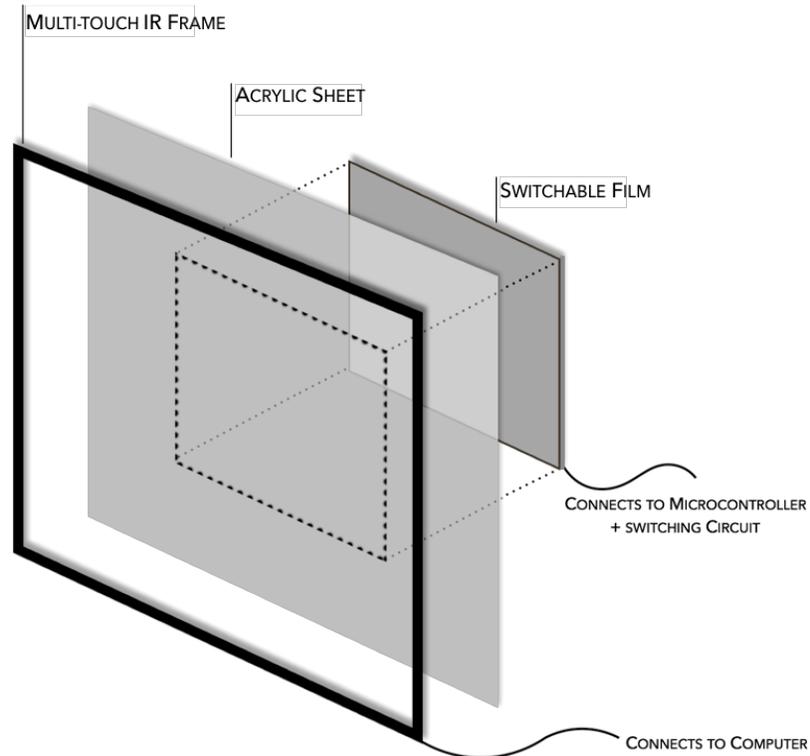


Figure 3.4: The door window assembly with multi-touch input, and the option to toggle between opaque and transparent states because of the switchable film.

3.1.2 Smart-Tint switching circuit and Hand Tracking System

To toggle between the transparent and opaque state of the new window assembly (Figure 3.4), I built a switching circuit using a solid state relay (SSR) [54] mated with an Arduino microcontroller [55]. The output of the SSR was connected to an extension cord, and I plugged in the switchable film to this extension cord. Thus, by sending a simple serial command from my computer, I was able to control the relay, in effect controlling whether the switchable film was transparent or opaque.

With hardware for multi-touch input and the window transparency control taken care of, a sensor to detect arm/hand position and posture was the only component left to figure out. The two viable options were the Microsoft Kinect [56] or the Leap Motion Controller [48]. The Kinect

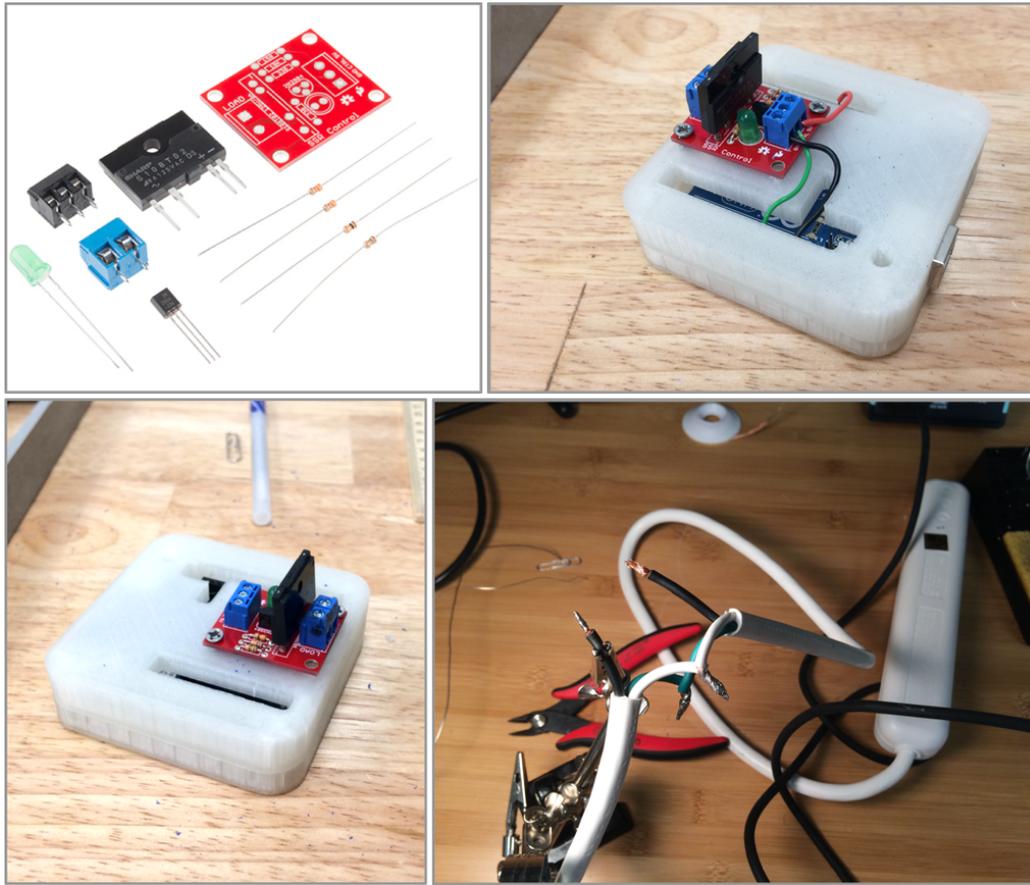


Figure 3.5: A solid state relay (SSR) and related electronic components (top-left). Arduino (top-right, inside a 3D printed case); the wiring was as follows: black to ground, the red wire to the Arduino’s 5V output, and the green wire to the Arduino’s digital input. A wire in the extension cord (bottom-right) was spliced and connected to the output of the SSR.

is more adept at tracking larger scale objects (e.g. body posture), and while it *can* detect arm or hand posture, the Leap Motion Controller is optimized to track hands and fingers (or stylus/pen like shapes). For AutoPoint, I did not need to track anything other than hand or finger movement; so, the Leap Motion Controller was my first choice.

Initially I mounted the controller on a wooden block that was taped to the arm rest in the door pad. Performance tests revealed that the sensor’s accuracy was rapidly dropping off halfway up the car window. It had to be mounted higher (Figure 3.6), and the results were quite promising (Figure 3.7).



Figure 3.6: The Leap Motion controller [48] was mounted on the arm rest, but that was too low. It was moved closer to the window sill for more accurate tracking.

To recap:

- *Problem statement* – I want multi-touch and in-air inputs for the purpose of selecting a Point of Interest (PoI). These inputs are to be given by the rear passenger.
- *What are those inputs* – point or touch from the car window.
- *What does the passenger point at* – at a target in the outside world, as seen from the car window.

The hardware having been designed to fit the above outline, the next step was the design of the software for the successful operation of the hardware.

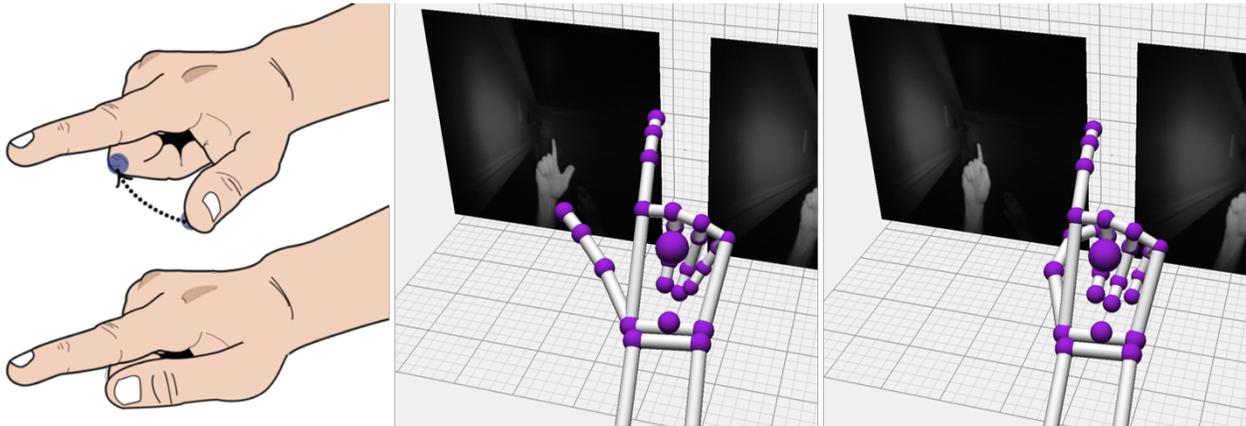


Figure 3.7: Visualization from the Leap Motion Controller with the SideTrigger [1] gesture.

The concept for the design of the software can be outlined as below:

- *External snapshot or freeze framing* - Isolating a frame or area of interest from the external world in motion (as seen from the car window).
- *Time Slicing* - Being able to bring an object or Point-of-Interest (PoI) that had already passed by, back into the passenger's viewport. This is the case when the initial snapshot could not or did not manage to capture what the passenger was interested in.
- *Area refinement and scene navigation* – What are the ways passengers could find a target that was wholly or partially obscured? Or, what if they just wanted to take a closer look at a target? Starting from the frozen snapshot, what could be the techniques that might be useful in isolating a target from a scene with many potential PoIs?
- *Target selection*– finally, to round off the design, how does one go about selecting the target (perhaps to get more contextual information)?

3.2 Software architecture and implementation(s)

There were two displays in the lab-setup (Figure 3.3, right): a) the 65" TV that shows/represents the external world, and b) the projected content on the car window. Each display was connected

to a computer, and the two computers were networked together in a client-server model to set up an interface between the external world data (i.e. content on the TV) and the data to be shown on the window (Figure 3.8). The software would have to provide the bridge between the server and the client, interpret the physical actions of the passenger and issue commands to the relevant hardware.

3.2.1 Pre-pilot software details

For the very first prototype – I call this the pre-pilot stage – I wrote two UWP (Universal Windows Platform [57]) applications written in C#, one ran on a computer connected to the TV (server), and the other on a computer connected to the projector (client). Both the computers were running

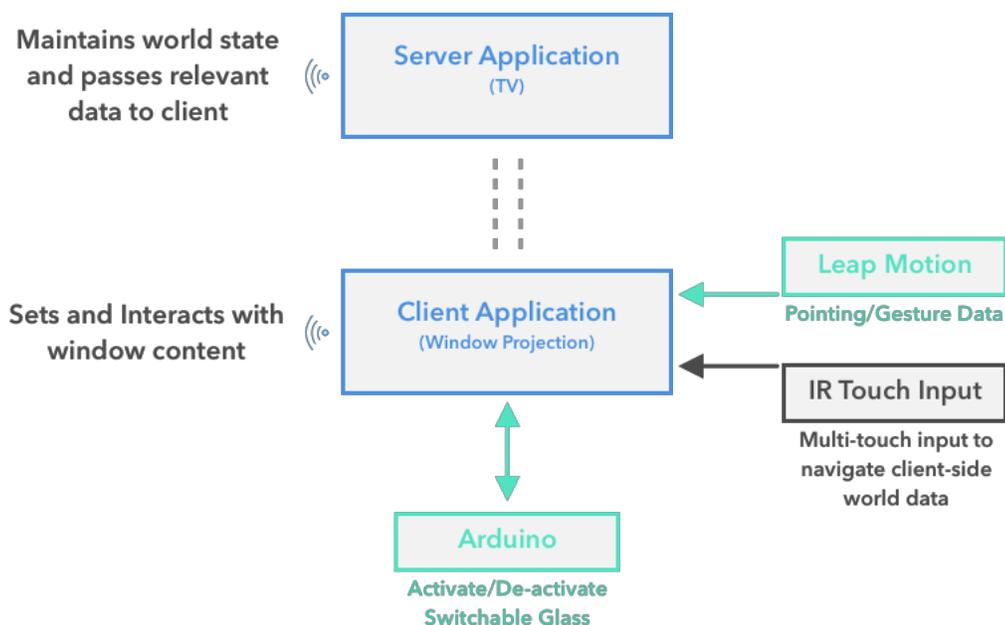


Figure 3.8: Software architecture for AutoPoint's lab setup. This architecture remained almost unchanged throughout the in-lab testing sessions, with changes limited to the visualization/UI layer only.

Windows 8.1. In essence, both the client and server applications were video players. The purpose of this prototype was to try out the basic interaction techniques with a handful of participants. This prototype had the following capabilities and features:

- The TV had a 10-minute video running on it. This video showed a scene recorded during a drive I took along Sheridan Road. In the lab test, the car window would have the switchable layer in its transparent state, i.e. the participant could see the scene playing on the TV, through the car window.
- A participant could tap the car window or perform the SideTrigger gesture to:
 - Make the switchable layer opaque.
 - Activate the car window projection. That is, instead of an all-black image, what is projected is a snapshot of the video playing on the TV at the instant the car window was tapped or the SideTrigger gesture (Figure 3.7) was performed. To clarify, if the participant tapped the window at 1 minute and 25 seconds into the video, the image projected on the car window would be the video preset at that timestamp.
- A participant could use a pan gesture (touch the car window with one-finger and move horizontally) to go back up to ten seconds in the video (rewind) to find a point in time that might be of interest. Note that, while the participant is interacting with the car window, the video continues to play on the TV (i.e. the external world is still moving relative to the car).
- Long-press the car window to make the switchable layer transparent again.

The purpose of this software prototype, with a pre-recorded video, was limited to checking for glaring design issues with this whole setup (hardware + software).

3.2.2 Testing pre-pilot software

Participants and testing parameters: recruitment for this pre-pilot stage was informal, and I had 3 undergraduate students between the ages of 19 – 21 (all male) stop by the lab to test this system and share their feedback. As a part of the protocol, I first gave them a demonstration of all the

**Table 1: Questionnaire data from PRE-PILOT session
(numerical data structured using 1 – 5 Likert scale)**

Questions	P0	P1	P2
Using pointing & in-air gesture to select an area of interest felt natural	2	5	4
Tapping the window to select an area of interest felt natural	4	5	5
Using the Pan gesture to find a time slice of interest felt natural.	4	5	5
Between the two, which did you prefer to use in order to select an area of interest (Point or Touch)	Touch	Touch	Touch
Comment	I couldn't tell if the sensor to detect pointing was working or not.	I am used to touchscreens and playing videos on my phone, so the touch interactions felt more comfy	Although pointing seems natural enough, the gesture seemed unnatural. Not sure if just being not used to it made me favor the touch condition
Other Comments and suggestions for improvements	A visual cue for pointing on the screen would have been nice	I liked the idea of seeing something and then being able to revisit it. I wish it went back more than a few seconds though.	I play video games, and to me the point and shoot gesture seems like something I could use with my PlayStation VR

features of the system. Special attention was paid to demonstrating the SideTrigger gesture, with the Leap Motion sensor visualization (Figure 3.6) shown as well. The participants were asked to practice it a few times till they reported that they were confident about performing the gesture. They used AutoPoint for a minimum of 15 minutes. In those 15 minutes they were asked to (and did) the following:

- Use touch input (single-tap) on the window to take an External Snapshot (ES), at least five times.
- Point towards the window and perform a SideTrigger gesture to take an ES, again, at least five times.
- Use the pan gesture on the window to go backwards or forwards in the video content now being projected on the screen. This was not a search task, I did not ask the participants to find something particular in the scene/video.
- Thereafter, make the window transparent by long-pressing/touching the car window.

Then, the three participants filled up a brief questionnaire (Appendix A: Pre-Pilot Survey) and shared their feedback (Table 1). The questions with a numeric value were structured using a 5-point Likert scale (1=strongly disagree to 5=strongly agree).

Lessons from the pre-pilot study: from observing the participants and getting their feedback, there were some obvious improvements that needed to be implemented:

- *Cursor Visualization for Leap Motion Controller:* This was already planned for later implementation. As it happened, all three participants were unsure of the correct wrist placement for the sensor to recognize the selection gesture. This lack of feedback also meant that they were tentative and overtly deliberate when pointing. Even though in this version I did not use the direction of the pointing gesture, some visual indication was necessary.
- *3D World:* The video player based implementation suffered from the shortcoming of being a 2D representation of a 3D world. With a lack of depth or height data in a pre-recorded video, hidden or occluded elements were eliminated from being potential targets for a participant. Also, interaction and visualization techniques were limited to manipulation of flat data. To explore scene navigation techniques in depth on the car window, the external world data had to be in 3D.

3.2.3 Pilot software details

The pre-pilot phase having served its purpose of identifying fundamental issues with the solution concept and its implementation, the next step was to implement a Pilot phase, incorporating the lessons from the pre-pilot, including the improvements mentioned above.

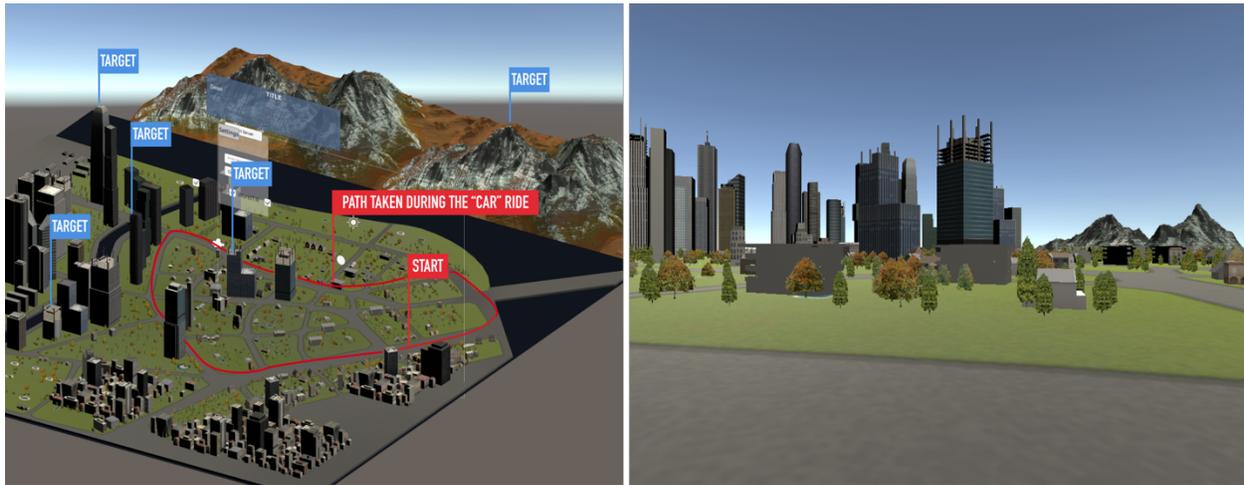


Figure 3.9: Bird's-eye view of the 3D world created in Unity (left), showing the path that the main in-game camera follows; the camera's view of world (in-car perspective) at the Start point of the path/loop (right).

Making a 3D External World

To create an external world in 3D, game engines were the obvious choice. I chose Unity [58] because it is free to use for non-commercial projects, and supports C# as a game scripting language. To create the external world, I placed buildings, mountains, and other elements in the 3D Unity environment. I was able to reuse parts of the code that I had already written for network connectivity, the serial communication with the microcontroller (to activate/deactivate the switchable layer), and the interface for the Leap Motion controller. The other modifications of the software are detailed below.

The goal was to create an abstraction of the experience of seeing a Point of Interest (PoI) through the car window and then being able to isolate and select that PoI from the external world. To that end, I made a cityscape with a few unique looking buildings and with mountains in the background. Certain elements in this cityscape were tagged as targets (or PoIs) for a participant to find and isolate from the rest of the scene.

I pre-defined a spline path around this cityscape and moved a main world camera along that path (Figure 3.9, left) for a pre-set time interval of 60 seconds. The choice to go with a spline path was deliberate. Instead of moving the camera around the world at a fixed speed, using spline interpolation gave a closer approximation of the characteristics of an actual car ride. In effect, from the user's point of view, the car was slowing down around tighter turns, and then speeding up on the straighter sections. The targets were placed in a way that at certain points around the path, they could be hidden or partially obstructed from view by other objects in the scene.

Inputs and interaction techniques

It is useful to frame the inputs and interaction techniques in terms of a task that a participant would try to complete. The task chosen is similar to the pre-pilot task, but different in subtle ways. Say the task is to select one of the peaks in the mountain range (Figure 3.9, right). The participant would:

- Look out of the car window and, on seeing the mountain, tap the window or perform a selection gesture (Figure 3.6), to take a snapshot of the 3D world. This snapshot is a part of the world, selected by the user, that could contain multiple targets (e.g. the mountain and buildings tagged as targets are in the same scene). I call this phase of the interaction: selecting an *Area of Interest* (AoI). This AoI (the snapshot) is projected onto the car window, and the user's attention is shifted from the TV screen showing the external world, to the interactive car window.
- Refine AoI: use smart phone like multi-touch gestures, like panning and pinch-to-zoom, to navigate around the AoI and get closer to the target.

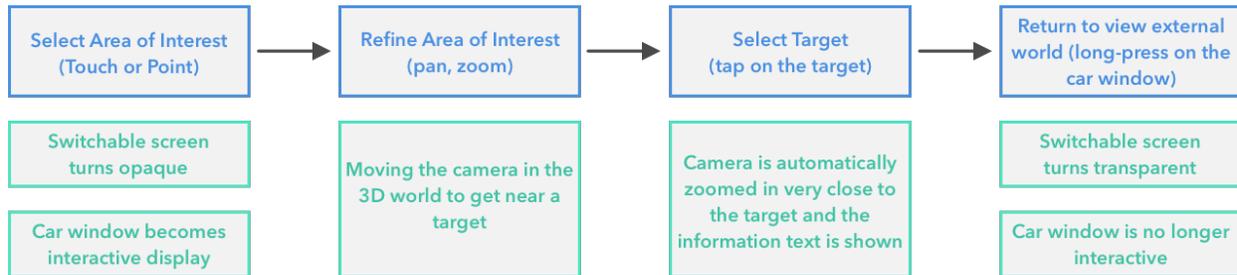


Figure 3.10: Interaction phases to select a target/PoI.

- Select target: tap the target (in this case the mountain) to select it. An information screen would show up (“You found an AutoPoint object, it’s a mountain!).
- Long-press anywhere on the window to make the switchable layer transparent again, thus shifting attention back to the TV screen.

The sequence of occurrences (Figure 3.10) starts with the passenger selecting an AoI. The gesture of selection is recognized by the IR touch frame or the Leap Motion Controller. The software is designed to command the SSR (via the Arduino micro-controller) to switch the screen into the opaque mode, whereupon the car window becomes an interactive display. Now, the passenger is able to refine the selected AoI; in the Unity world, this represents moving the camera around. The remainder of Figure 3.10: Interaction phases to select a target/PoI. is self-explanatory.

There is an important edge-case that I implemented as well. It is possible for a user to tap or point + gesture at the target on the TV and *directly select* it. The rationale was simple, during an actual car ride, the PoIs can be far away from each other, so that a passenger ought to be able to select a target directly without having to go through the refinement step. I outline this and some other implementation details in the next section.

Implementation details

The underlying software architecture was still a client-server model unchanged from the pre-pilot stage. The computer powering the TV displaying the external world, was the server, while the computer powering the car window display (i.e. projector) was the client. When the user touched the window or used the point + gesture input, I took the camera's position and orientation at that instant from the server and passed it on to the client. Since the client and the server shared the model of the world, taking a snapshot or selecting an AoI was, effectively, a matter of setting the client's camera to the same position and orientation as that of the server.

Selecting an area of interest and targets: It helped that the Pilot version of AutoPoint's software was built on top of a game engine. It was quite simple to cast a ray from a point of origin, give it a direction vector, and check against all the targets that had colliders in the scene. The origin and direction of the ray was different between the touch and pointing inputs. For touch input, taking the touch coordinate as the point of origin, and setting the direction as the camera's normal was the obvious solution, and it worked well. For pointing, the origin of the ray was the tip of the

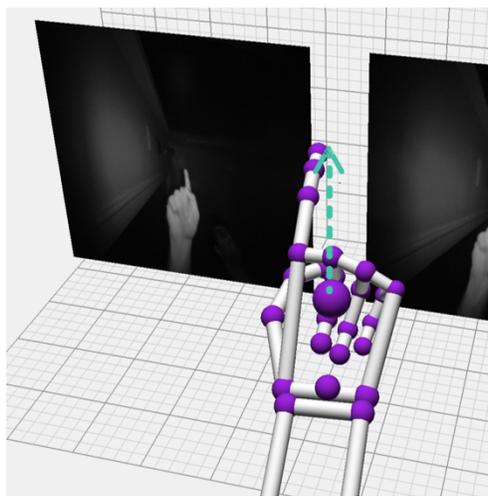


Figure 3.11: A vector from the wrist to the top of the index finger was used to set the direction of a ray.

index finger, but the direction vector was computed based on the index finger's position relative to the wrist (Figure 3.11).

This subtle difference in the implementation affected both the selection of an area of interest (AoI), and the direct selection of a target (i.e. without having to go through the AoI refinement phase). For selecting an AoI with the point + gesture input, the area chosen by AutoPoint would usually be offset in the direction being pointed at when compared to the touch input. The same would be true for selecting a target (PoI).

In *Pointing as Situated Practice* [9], Goodwin writes that pointing gestures are a rich source of information and embody a “domain of scrutiny”. How does AutoPoint's point + gesture technique relate to Goodwin's domain of scrutiny? Specifically, the question is: does point + gesture improve the domain (the AoI) identification, as compared to point alone? Another question I try to find the answer to is: how convenient will a user find the point + gesture technique as compared to, say, the touch technique?

Cursor visualization while pointing: To improve the point + gesture experience, I designed a semi-transparent cursor that showed up on the car window when the participants positioned their wrists above the motion sensor and pointed at the TV (external world). This was fairly easy to do because Leap Motion's API has a simple way to map the 3D coordinates of hand movements into 2D screen coordinates [59]. The cursor provided participants with visual feedback of when the motion sensor was actively tracking their fingers.

3.2.4 Pilot Testing

As with the pre-pilot study, the numeric data in the questionnaire were structured using a 5-point Likert scale (1 = strongly disagree and 5 = strongly agree). At this stage of AutoPoint's development, there was no monetary remuneration for the participants.

In a real-world scenario, users would be free to explore and select an element that piqued their interest. Since that is not feasible for a lab study, I handed participants a printout with photos of select 3D-world elements (a mix of buildings and mountains) that were targets. Any of these targets (PoIs), when selected by a tap or a touch, would simultaneously trigger the camera to zoom and a textbox to appear, with more information about the target.

The participants were asked to do trial runs till they reported that they could use all the interaction techniques satisfactorily, that is, touch or point + gesture to select AoI, followed by pan and zoom to refine AoI and finally, tap the target. The point + gesture to select an AoI took more trial runs than the touch condition.

Participants were instructed to select a minimum of 7 targets each for the touch and the point + gesture input techniques. I informed the participants that the goal was to capture user experience and not how quickly they finished the task. Including the trial runs, each session took between 15 – 20 minutes.

For quantitative data points, I logged the timestamp of each successful target selection, the type (pan or zoom) and timestamps of interactions that each led to a target, and the number of errors. I define an error as the case when a participant either touched, or used point + gesture to select an AoI, but then did not continue on to perform a successful target selection. In the pre-pilot study, I had observed that participants would sometimes accidentally touch the touch screen, or the Leap Motion sensor would get triggered because of a momentary loss of tracking; I kept track of the number of instances in which that happened.

Participant feedback and discussion:

Even though I logged some performance / quantitative measures, the goal of this Pilot stage was still to iron out design and implementation weaknesses and flaws, hence to evaluate qualitative

Questions	P0	P1	P2	P3	P4
Using pointing & in-air gesture to select an area of interest felt natural	2	3	4	3	4
Comment	Pointing was fine, but the gesture was odd.				
Using touch gestures to select an area of interest felt natural	3	5	4	4	5
Using touch gestures to isolate a target felt natural	4	5	4	5	4
Between the two, which did you prefer to use in order to select an area of interest	Touch	Touch	Touch	Touch	Touch
Comment	The pointer was "jiggly", and I had trouble lining it up	I liked being able to "stop time", the point and shoot technique was interesting and found that to be more playful	The gesture took getting used to, but thereafter felt fine. It seemed fairly accurate as well. However, touch was just more accurate	The pointing felt unreliable, but the visual cue was quite nice/helpful.	I liked being able to select a target by pointing and gesturing. I had to move my head less and preemptively select something. It is only because I had to gesture multiple times did I rate it lower than touch
The system can encourage exploration of unfamiliar areas.	5	5	4	4	5
Other Comments and suggestions for improvement	Making the pointer size bigger would make it easier to use or focus on. Transition from point to touch felt unnecessary	The ergonomics seemed a little off. I felt like I needed to twist my neck to look through the window I also kept looking at your computer once in a while	Seems similar to a heads-up display for the window. It'll be nice to see the speed the car was traveling at as well.	N/A	I can see this being used in a car but perhaps try to make the pointing more reliable

characteristics. Participant feedback is summed up in Table 2 and Table 3 has the number of errors for each technique used to select an AoI. The totals are out of a possible 35 for each technique.

All five participants preferred the touch input over the point + gesture input for the selection of the AoI. Since the sample size is quite small, running a Mann-Whitney *U*-test on the Likert scale

data would not reveal anything of substance. While the same is generally true for simpler statistical measures such as mean and median, the scores for how natural the interactions felt (touch: mean = 4, median = 4; touch + gesture: mean = 3.2, median = 3) do show that touch tends to have a higher rating. While this could be because of people being more used to multi-touch interactions on smartphones, I am not making a strong claim based on these numbers (and I also acknowledge that calculating mean for ordinal data is sometimes frowned upon [22]), but the higher numbers for touch input fits the general feedback recorded in Table 2.

Table 3: Number of errors for the two input techniques that are used initially to take a snapshot of the external world (AoI).

Participant	Touch	Point + Gesture
P0	0	3
P1	1	3
P2	1	1
P3	1	2
P4	1	1
Total	4	10

Most of the participants had comments about the point + gesture input, some being critical of the reliability. This criticism is reinforced by my recording of the total number of errors (Table 3) for each input condition; point + gesture had more than double the number of errors (10) compared to touch (4).

Based on my observation – also echoed by participant P4 – this could be attributed to the design and implementation of the gesture. Firstly, there is a variability in the Leap Motion Controller’s model of the hand that makes threshold and distance-based triggers tricky to implement. Thus, if I simply measure the distance of the thumb from the curled middle finger (Figure 3.7) and say that a command should be triggered when distance < X, there would be false positives (due to a change in the model of the hand between two pointing gestures). Secondly,

participants stop paying attention to the gesture and start performing it less deliberately after a couple of successful AoI selections. To make matters worse, there is an interplay between these two factors. That is, if the threshold for a successful gesture is increased or decreased significantly so as to accommodate a wider margin of error to compensate for the sensor's variability, it becomes incumbent on the participants to be *more* deliberate, when their natural instinct is to be less so (hence, P4's repeated failures with the gesture).

Besides that, there were some positive comments about the visual cue (transparent cursor) that I had added for this pilot version of the software (based on feedback from the pre-pilot version). That being said, all the participants tended to wait for a split second for the cursor to show up before trying to perform the gesture. I wondered if that visual clue would be truly necessary if the participant were familiar with the working of an accurate in-air gestural system. This is so because, for the duration of the pause, a participant stopped focusing on the external world (on the TV), and focused on the window, before focusing back on the external world again. That seems like a needless artifact of this particular system's design. Even though point + gesture had issues with errors and false positives, some of the feedback from the participants was encouraging enough to develop it further. In the next phase, I planned on adding a continuous calibration step before the gesture recognition to see if that led to lower error rates and higher satisfaction.

Since I had replaced the video recording based system to a 3D world, I asked if participants felt curious to explore this world with pan and zoom gestures on the window. Having specified the permitted targets, I could not facilitate an exploration mode in the task I presented to the participants, but most participants seemed to like the idea of such a facility (mean = 4.6, median = 5). Again, I report this just as an indicator and not a major claim. I continued with this question in

the next phase of the project (31 participants in Phase II), at the end of which the sample size would be large enough to draw meaningful conclusions.

Based on my own observations and verbal feedback from the participants, I intended to make some changes on the hardware front for the next phase, namely: a) replacing the acrylic sheet in the window display with glass because the acrylic sheet's friction properties were annoying to a couple of participants, b) a bigger window display that would be more immersive for a participant during the area refinement phase, and c) a way to ensure that participants did not try to peer around the edges of the car door.

Finally, to round off this chapter, I would like to mention a thing of note that came out of this phase. Direct target selection (i.e. being able to select a target right away by touch, or by point + gesture, without going through a target refinement phase) actually hindered the experience of selecting a target. All the errors in the touch condition were due to participants accidentally tapping a target when what they really intended was to select an AoI. This meant that they had to clear the window by long-pressing and then attempt the target selection again, a fairly frustrating experience, especially if the target was no longer in the field of view. In a context where the world is in motion, interactions benefit from having a level of coarse-ness or in-built fault tolerance. Being able to take a snapshot in time to get an AoI (coarse interaction), and then being able refine it seems to strike the right balance between performance and error tolerance. Like P1 said, "I liked being able to stop time".

CHAPTER 4

AutoPoint Phase - II

Based on the testing and feedback from Phase-I, I refined the hardware and made significant additions to the software. For the hardware, I decided to build a larger viewing frame and to replace the acrylic sheet with glass. At the same time, I wanted to cover the test setup with a canopy to mimic, roughly, the inside of a car and to prevent users from indulging the occasional temptation to peek around the car door. For the software, the focus was on adding touch-input based capabilities to the area refinement stage of AutoPoint (Figure 3.10), with the object of enriching the overall experience. In addition to the pan and zoom interactions, I developed a number of other touch-input based interactions and visualization techniques – like depth-of-field, world tilt, and time slicing (described later) – and tested them with a total of 31 participants.

4.1 Hardware Refinements

The building blocks of the hardware stayed the same as in the Pilot phase, but I made a number of tweaks that were designed to improve the user experience.

Changes to the window assembly

I replaced the acrylic sheet (Figure 3.4) that covered the IR touch-frame with a glass panel. Two of the five participants in the Pilot phase mentioned that swiping gestures (i.e. both pan and zoom) felt less than ideal because their fingers tended to “stick” to the acrylic surface. I think that the acrylic sheet caused a bit of static build-up that was responsible for the perceived stickiness. While the glass panel had better friction properties for this application, it was also significantly heavier than the acrylic sheet. I had used double-sided mounting tape to install the window assembly



Figure 4.1: Acrylic layer replaced with glass; it needed a sturdier aluminium mount (top-left, right), the switchable film now spanned the entire window (right), and the projector was mounted on a tripod by repurposing a plastic board.

during Phase-I, but that was not secure enough for the new assembly. An aluminium bracket (Figure 4.1) served as the mounting plate for this hardware iteration.

The switchable film (Figure 3.4) that covered only a section of the window was also replaced by a film that spanned the entire width and height of the IR touch-frame (i.e. the full window).

In Phase-I, I had assumed that the smaller switchable film (which served as the display / projection surface) would be a better design choice. While the participants could choose to focus on the area covered by the switchable film, they would always be able to see the external world in motion in their peripheral vision. For the same reason, I had also thought that after selecting a target, returning to an unobstructed view of the external world would be a less jarring experience.

But feedback from almost everyone during the pre-pilot and pilot phases informed me that they found the motion in the periphery of vision to be quite distracting.



Figure 4.2: Tent poles connected to studio tripods (left) formed a structure that was used to drape a cloth over the car door to make a canopy (right). The window display was significantly larger than the prior iteration.

Projector mount

A minor change to the hardware setup included the projector mount. I fashioned a tripod mounting plate out of an old plastic board to make it easier to set up the projection system (Figure 4.1, bottom-left) for each test session. This also enabled me to make quick adjustments to the height of the projector and reduce the probability of glare for particularly tall participants.

Cloth Canopy

Feedback and observations from two participants in Phase-I led to this addition. I noticed that a tall participant would peer over the door frame once in a while. Another participant mentioned that she tended to glance at the computer I was using to drive the TV display. Both of these issues were taken care of by affixing a couple of tent poles (Figure 4.2, left) that I could drape a cloth over

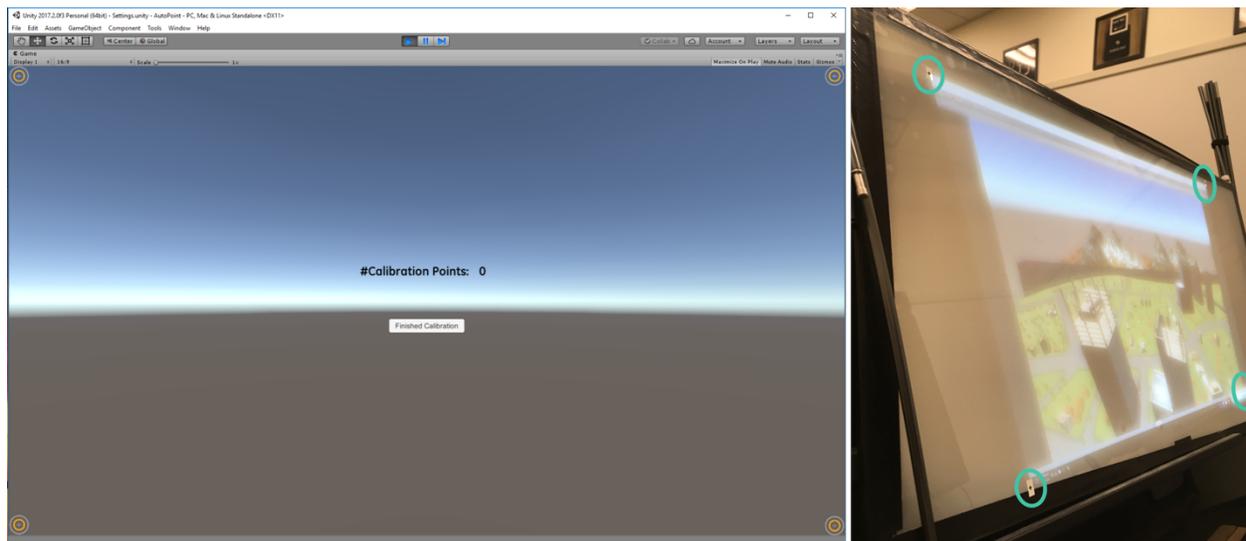


Figure 4.3: A 4-point calibration step mapped the range of IR touch-frame coordinates to the projected display bounds.

to create a canopy (Figure 4.2, left). A side effect of this canopy was fairly obvious, the projection-based car window display looked and felt a lot brighter and sharper. Overall, this simple change made the hardware setup a much better approximation of a car interior.

4.2 Software changes and refinements

With a change in the hardware, there was an additional calibration step added to the AutoPoint system. However, with this iteration of AutoPoint, I was more interested in implementing a number of multi-touch interactions that could be useful in the area refinement phase (Figure 3.10) and examine if they performed better (or were deemed to be more useful by the participants) than the basic pan and zoom techniques from the previous iteration(s). Many of these techniques were designed on the basis of previous research on 3D content navigation [8,10,24,34,35,44] that I have covered in the chapter Related Work.

Touch Calibration Step

I was using a combination of the lens tilt and keystone controls on the projector to project an image onto the window assembly from a short distance and a steep angle. Since the projected

image was distorted, the range of coordinates received from the IR touch-frame had to be mapped to the range of coordinates spanned by the image on the display (Figure 4.3).

New Interaction techniques for area refinement

With AutoPoint, I wished to investigate the limitations and successes of multi-touch and in-air gestural inputs when a 2D interactive area is all that is available to observe or learn about an external 3D world in motion. However, up to Phase I, I had looked at only *two* touch-based interaction techniques on the window. For Phase II of the in-lab study, I added an additional four touch-based interaction techniques to the pre-existing pan and zoom techniques from the previous iteration. All of these interactions were applicable during the area refinement phase (Figure 3.10).

- ***Rotate:*** The simplest of the four additional techniques, it was fairly trivial to add a rotate gesture to the existing pan and zoom gestures. This was added mostly to ensure that AutoPoint supported the fundamental set of interactions that participants would be used to with smartphones. That being said, I did not expect this to be particularly useful in the manipulation of a 3D world; this was implemented identically to how a photo is rotated on a smartphone screen.
- ***Depth-of-Field:*** In the real world there can be many PoIs in an area a passenger travels through and, depending on the surrounding areas, the density of PoIs can be quite high (e.g. buildings in a large city like Chicago). In the 3D world I created for the lab study (meant to be an abstraction of the real-world), there were many instances where multiple targets would be visible from the window while the camera moved along the pre-determined path (Figure 3.9, right). To pick one out of a jumble of PoIs, a method was needed to unclutter the image. Hata et al. [11] have reported on the effectiveness of background blur as a technique to draw attention to a PoI in a 3D environment. It is

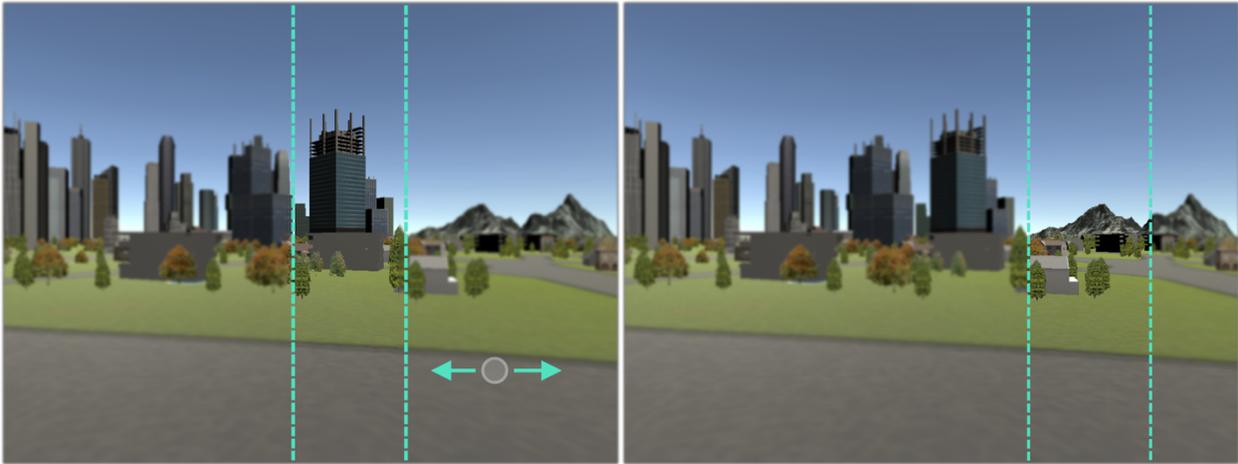


Figure 4.4: A depth-of-field effect was implemented by blurring out everything in the scene other than the target (bluish grey building) closest to the camera (left). An area of focus (a bounding box) could be moved by a participant to jump to the next PoI (the mountain peak) by performing a pan gesture on the window.

interesting to note that this technique is well known to photographers; bokeh or the bokeh effect [60] has been used for years to draw attention to a subject in photos for years.

I implemented a depth-of-field effect akin to the background blur referred to above. When a participant took a snapshot of the external world (with touch, or with point + gesture), I calculated which target was closest to the camera, and then applied a blur outside the target's bounding box (Figure 4.4). Participants could choose to pan the scene and move this area of focus to other targets if they wished to.

- **World Tilt:** The implementation of this interaction technique was based on UniCam's [44] design. This was an orbiting technique [34] where the camera height and tilt were being changed simultaneously. The camera transforms were calculated by mapping the 2D movement on the IR touch-frame over the surface of an invisible sphere of radius equal to the distance of the camera from the center of the scene. Since the visual effect of this was similar to tilting the world, or getting a bird's eye view (Figure 4.5), I called it World Tilt.

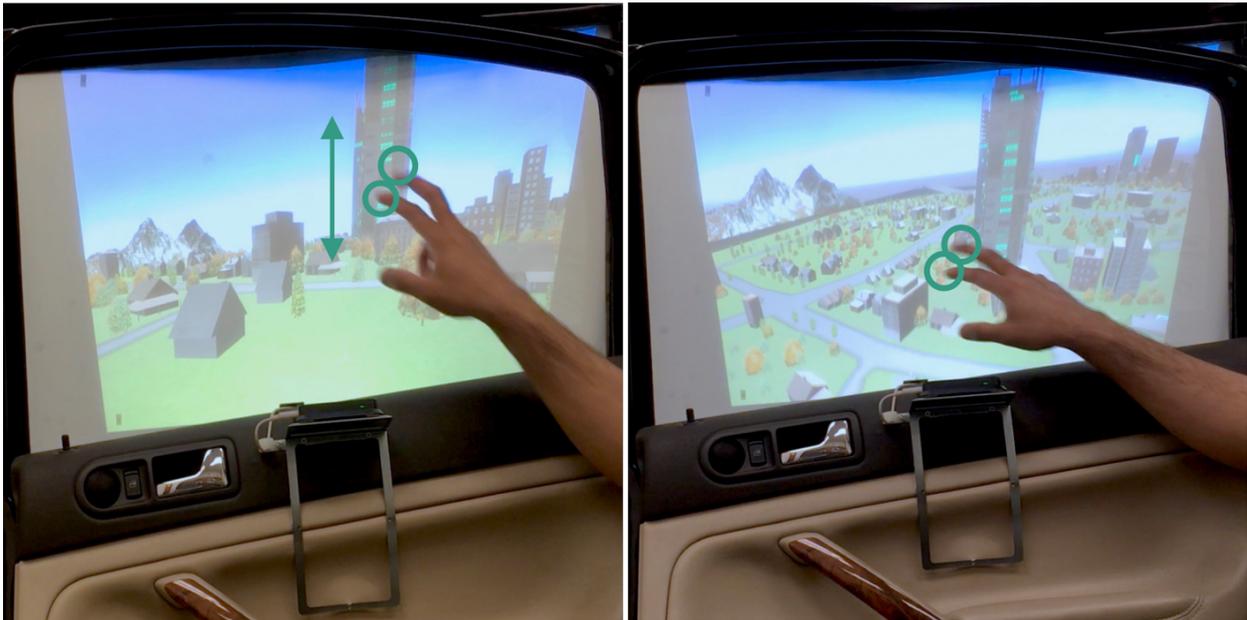


Figure 4.5: World Tilt: moving two fingers vertically on the screen gives the participant a bird's eye view of the scene.

While World Tilt could have been effected by means of a single touch-point (i.e., by using just one fingertip), it seemed more appropriate to allow participants to use this interaction technique and the pan gesture without having to switch modes by tapping an onscreen button or a widget. In other words, the choice of the two-finger input was for the purpose of enabling a seamless transition from the pan mode to the World Tilt mode.

It was simple to disambiguate this gesture from a pinch-to-zoom gesture (which also requires two points of contact) by looking at the initial configuration of the two touch points. If the two touch points start out far away from each other (distance > threshold) or they have a larger height difference ($Y\text{-coordinate of } 1^{\text{st}}\text{touch point} - Y\text{-coordinate of } 2^{\text{nd}}\text{touch point} > \text{threshold}$), then the gesture is interpreted as a zoom input; otherwise the gesture is interpreted as a World Tilt.

The main purpose of World Tilt was to reveal PoIs occluded by other elements in the world. For example, on a drive next to a forest, World Tilt could reveal a lake somewhat

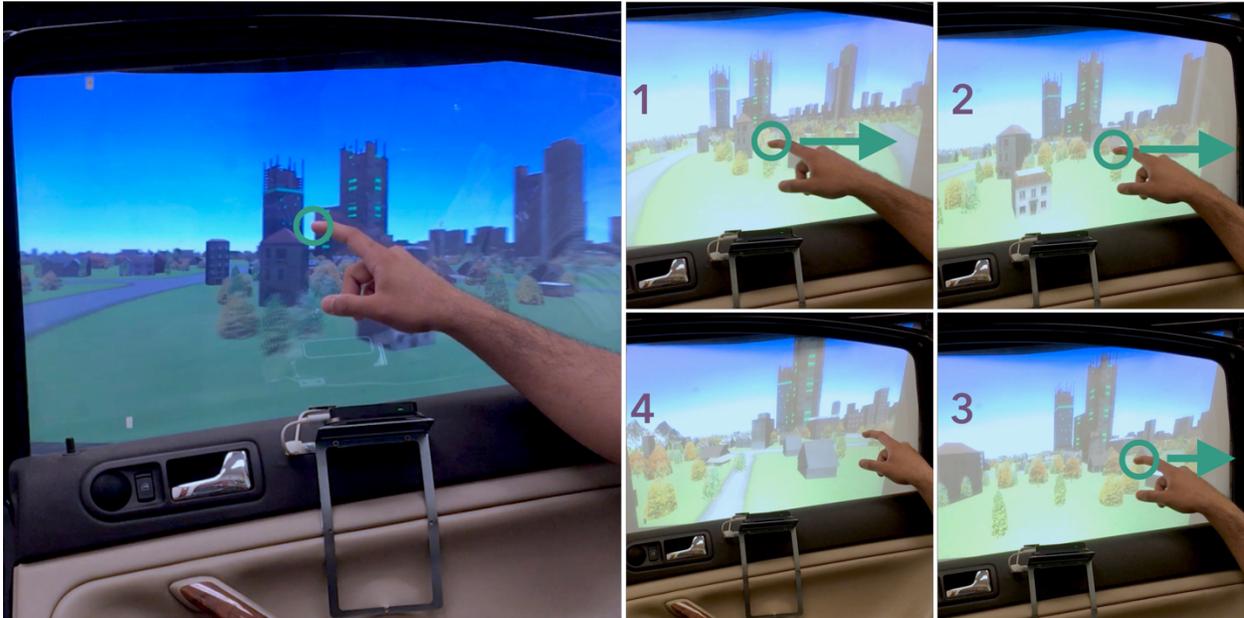


Figure 4.6: Time Slicing: Take a snapshot of the external world by tapping the window (left). Then swipe anywhere on the screen to go backwards in time (right). In this case, the mountain had passed by already (1), but Time Slicing allowed it to be brought back into the frame (4).

close to the road that was hidden from view; or a restaurant that was hidden by a tall building, etc. In addition, this technique offered a good perspective of the lay of the land, and how spaces were connected to each other.

- **Time Slicing:** Depending on how close a PoI was to a car, or how fast the car was moving (or a combination of the two), passengers might pass by a PoI before they got a chance to engage with it.

To tackle this scenario in the lab setting, I implemented a technique that enabled a participant to revisit a section of the path shown on the TV screen. AutoPoint’s client-server software architecture made this fairly easy to do. I used a circular buffer on the server side to store the camera’s position and orientation. Depending on the size of the circular buffer, I had historical data – in this case up to a thousand frames – that the participants could use to travel back in time, i.e. retrace their steps. When participants took

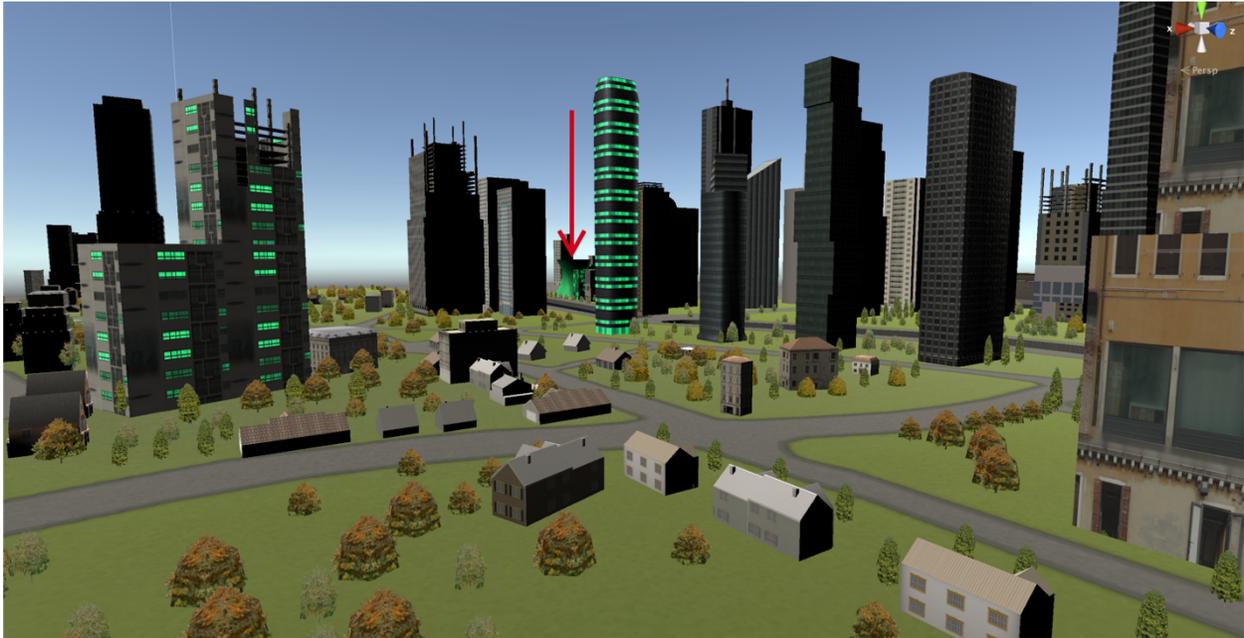


Figure 4.7: The target buildings now had a texture with a green highlight. I added smaller targets (red arrow) that could be easily occluded by other elements in the scene.

a snapshot of the external world, using touch or point + gesture, the circular buffer's contents were sent to the client (i.e. the window display). Participants could then swipe on the window to set the client-side camera's position and orientation to a value stored in the circular buffer, in effect going backwards or forwards in time (or Time Slices, if you will). Figure 4.6 shows a few stages of using Time Slicing to revisit a target that had already passed by.

With these four additional techniques, AutoPoint supported a total of six different interactions, namely, pan, zoom, rotate, depth-of-field, world tilt, and time slicing.

Changed target appearance

One last change to the software was mostly cosmetic. In Phase-I, I had shown printouts of targets to participants. Even though practice made it easy enough for them to recall which elements were targets, for Phase-II, I changed the texture on target buildings to include a green highlight (the

mountains remained unchanged). This also allowed me to scatter some smaller targets around the world and observe how many participants selected a target that was easily obscured (Figure 4.7).

4.3 Testing with Participants

Participants were recruited by posting flyers around campus (Appendix E:) and they were paid \$15 for each session. I was able to recruit a total of 31 participants (14 male, and 17 female) between the ages of 19 – 31. Since there were two techniques to select an Area of Interest (touch, or point + gesture), and six different area refinement techniques (pan, zoom, rotate, depth-of-field, world tilt, and time slicing), asking participants to select 7 targets for each condition would have resulted in very long test sessions. To make each test session shorter, I chose to divide this round of testing into two distinct sets.

4.3.1 Test Version 1 (V1)

Participants and testing parameters

Version 1 (v1)

Take External Snapshot	Area Refinement Phase	# Targets to Select
Touch	Pan + Zoom + Rotate	5
	World Tilt	5
	Depth-of-Field	5
Point + Gesture	Pan + Zoom + Rotate	5
	World Tilt	5
	Depth-of-Field	5
Total		30

Figure 4.8: The various combinations tested in V1 with each of the seven participants.

This set included all the area refinement techniques excepting time slicing, and the number of targets that a participant had to select was reduced to 5 per combination of techniques (Figure 4.8). Notice that the pan, zoom and rotate are grouped together. This was done to make a distinct subset of interaction techniques that participants were already familiar with. In addition, World Tilt and

Depth-of-Field were both conditions where the pan + zoom + rotate continued to be available for use. In the case of the Depth-of-Field visualization, a participant had to tap a button to enable or disable the pan, zoom, and rotate operations because both Depth-of-Field and pan use a single touch point for scene manipulation. No such button was needed for World Tilt because it used a two-finger pan gesture.

The goal was to record user experience, and then shortlist the techniques that worked well in V1 for more thorough testing in Version 2 (V2). Seven participants tested this system. Each test session took between 30 – 35 minutes, including the time taken by the participants to fill out a questionnaire (Appendix C:). It was mostly similar to a Phase-I questionnaire. A question that I would like to highlight, is the one where I asked the participants to rank the various interaction techniques. These data and my observation of the test sessions would help inform the parameters for testing in V2.

Similar to Phase-I of the development of AutoPoint, the participants were asked to do trial runs till they reported that they could use all the interaction techniques satisfactorily. I logged the timestamp of each successful target selection, the type and timestamps of interactions that each led to a target, and the number of errors. Just as in Phase-I, I define an error as the case when a participant either touched, or used point + gesture to select an AoI, but then did not continue on to perform a successful target selection.

Participant feedback and discussion:

Among the new interaction techniques that could be used to refine an AoI, World Tilt seemed to perform particularly well. I parsed the log data to check whether participants used the pan + zoom gestures in conjunction with World Tilt to refine an area with a target. From a total of 70 target selections (2 input techniques to select AoI x 5 successful target selections per input technique x 7

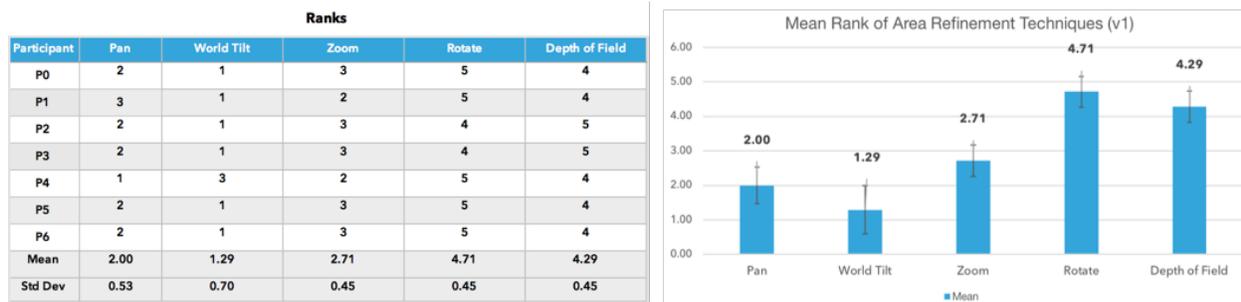


Figure 4.9: V1: Ranks assigned in order of preference to interaction techniques used to refine an Area of Interest. Lower scores are better (rank = 1 is the best, rank = 5 is the poorest).

participants) using World Tilt, only 13 (~19%) also needed to make use of the pan, zoom or rotate operations. Out of the seven participants, six ranked it the highest (Figure 4.9) as well, and made mention of the fact that the bird's-eye-view was a perspective that was new and of interest to them (Table 5). Somewhat encouragingly, some participants did end up selecting the smaller, partially occluded targets (a cumulative total of 9 times) that I had scattered around the 3D world, and all of those interactions involved World Tilt (and some of them involved the pan gesture as well).

Participant	Touch	Point + Gesture
P0	0	3
P1	2	2
P2	0	7
P3	1	2
P4	2	2
P5	0	3
P6	2	3
Total	7	22

Table 4: V1, number of errors for the two input techniques that are used to take a snapshot of the external world (AoI).

The Depth-of-Field interaction/visualization performed surprisingly poorly. Most participants chose to tap the button that enabled the pan, zoom, and rotate operations, and then use those to refine the area of interest. The log data showed that participants exclusively used the Depth-of-Field interaction for only 11 out of a total of 70 potential target selections. Correlating this with the

Table 5: Questionnaire data from V1 session. The question that asked participants to rank the interaction techniques, has been pulled from this table and reported separately.

(Numerical data structured using 1 – 5 Likert scale)

Questions	P0	P1	P2	P3	P4	P5	P6
Using pointing & in-air gesture to select an area of interest felt natural	3	3	3	3	4	4	4
Using touch input to select an area of interest felt natural	5	5	5	4	4	4	5
Using touch gestures to isolate a target felt natural	4	5	5	4	4	5	4
Between the two, which did you prefer to use in order to select an area of interest	Touch	Touch	Touch	Touch	Touch	Touch	Touch
Comment	If you tested the pointing from further away, then course, that'd be the only option and hence useful, but otherwise, the touch felt way better		I made more mistakes with the pointing.				
The system allowed you to isolate a target easily	5	5	5	5	4	4	5
Other Comments and suggestions for improvements	I would totally use world tilt to explore areas around me. I love to stare at maps, so this would be just right for me.	DoF would be useful if I were taking a photo, but otherwise it just got in the way	I accidentally activated the system a few times, that was frustrating. But other than that I loved many of the other ideas. Especially world tilt, we do not usually get to see that perspective	I get motion sickness if I stare at a phone when I am traveling in a car, and prefer to stare outside. I wonder if this kind of a system would be okay for me or just make matters worse	Rotate seems to not be very useful, I only used it accidentally, and then had trouble setting the world right. Similarly, World Tilt was harder to do than pan and zoom.		If I were taking a photo of the world outside, rotate might come handy to correct the angle later, but otherwise, World tilt is sort of a rotate that I like a lot more.

data from the questionnaire does seem to reveal an issue. Firstly, this technique was ranked – along with rotate – at the bottom of the pile by nearly all participants. Secondly, a participant mentioned that this technique got in the way of target selection. This was interesting to me.

I had observed that participants tended to take a snapshot of the external world as soon as they saw a target appear within their field of view, despite my making it clear that I was not measuring how quickly they selected a target. The fairly rudimentary implementation of the Depth-of-Field technique meant that the target closest to the camera would be in focus, but in any frame with multiple targets there was a chance that the user intended to select a target different from the one closest to the camera. With this mismatch between user intent and the default target chosen by the system, most users tended to take a snapshot and then *immediately* try to override the default by tapping the button for the pan and zoom technique.).

As in Phase-I, some participants were critical of the point + gesture input. This criticism is buttressed by my recording of the total number of errors (Table 4) for each input condition; point + gesture had more than triple the number of errors (22) compared to touch (7). That being said, I had been successful in making the point + gesture slightly more reliable than it was in Phase-I with close to a 10⁰% drop in error rate that was brought about by changing some of the parameters of the gesture detection (this included using multiple timers to track hand states over a period of time).

The above were the lessons from the V1 test sessions. The most important of these lessons was the value of World Tilt. Hence, World Tilt was the most obvious interaction technique to be carried over from V1 to test session V2. I decided to persist with the point + gesture for V2 as well.

4.3.2 Test Version 2 (V2)

Participants and testing parameters

In this set, there were three interaction techniques in focus, namely, World Tilt (carried over from V1), Time Slice, and Pan + Zoom. I removed rotate as an option based on the feedback from V1. Since Time Slice used a single touch point to work, a participant had to hit a button to enable Pan + Zoom (simultaneously disabling Time Slice) if they so desired. This action of pressing a button was similar to the implementation of the Depth-of-Field option from V1.

Version 2 (v2)

	Take External Snapshot	Area Refinement Phase	# Targets to Select
Touch		Pan + Zoom	3
		World Tilt	3
		Time Slice	3
Point + Gesture		Pan + Zoom	3
		World Tilt	3
		Time Slice	3
3 different durations (45, 60, 90 seconds)	Participant's Choice	Participant's Choice	15 (5 per duration)
Total			33

Figure 4.10: The various combinations (per participant) tested in V2 with 24 participants.

The combinations tested in V2 are shown in Figure 4.10. Note that I reduced the number of targets that needed to be selected from the 5 in V1 to 3 here for both the touch, or point + gesture input techniques. This decrease was in recognition of the addition of another condition to the experiment (Figure 4.10, highlighted in orange), and I wished to keep the duration of a test session within reason. In this new condition, participants were free to choose any combination of input techniques to select a target. I also varied the duration of a loop (Figure 3.9) between 45, 60, or 90 seconds, the duration being a proxy for how fast a car was traveling. As in V1, I was logging the sequence of events that led to the selection of a target, and this log would reveal whether the

duration of the loop (i.e. the car's speed) had an effect on the kind of interaction techniques participants used (e.g. did they use Time Slice more frequently for when the car was going fast?).

Allowing the participants to choose a technique (touch, or point + gesture) to take a snapshot of the external world necessitated some tinkering with the software. Since all the prior tests had shown that point + gesture had higher error rates, I had to ensure that the Leap Motion sensor would not record a false positive for the point + gesture, when the participant's intent was to touch the window to take a snapshot. I adopted the simplest and fastest way to get around this issue: I carefully observed the participants, and if I saw them performing the gesture, hit a key on a bluetooth keyboard to trigger the external snapshot (a Wizard of Oz implementation [15]). The participants were not informed about this switchover from the Leap Motion sensor based trigger for the external snapshot that they were using initially, to my manual intervention during this section of the test session, and no one reported any change in the user experience.

Twenty four participants tested this set and each test session took approximately 40 minutes which included the time taken to fill out a post session questionnaire (Appendix F). The questionnaire and the test protocol was consistent with V1. This time too, I asked the participants to rank the various interaction techniques. The participants were asked to do trial runs till they reported that they could use all the interaction techniques satisfactorily. I logged the timestamp of each successful target selection, the type and timestamps of interactions that each led to a target, and the number of errors.

After a test session, and after the participants had filled up the questionnaire, I did a quick debriefing session with them that was audio recorded. During the debriefing I asked them to express their general thoughts about AutoPoint, any potential applications that they could think

Participant	Ranks		
	Pan + Zoom	World Tilt	Time Slice
P0	1	2	3
P1	2	3	1
P2	1	3	2
P3	3	2	1
P4	2	3	1
P5	2	3	1
P6	3	1	2
P7	3	2	1
P8	3	2	1
P9	3	1	2
P10	2	1	3
P11	3	2	1
P12	3	1	2
P13	3	1	2
P14	3	1	2
P15	3	2	1
P16	3	1	2
P17	2	3	1
P18	3	2	1
P19	3	1	2
P20	2	3	1
P21	3	2	1
P22	1	3	2
P23	3	2	1
Mean	2.50	1.96	1.54
Std Dev	0.72	0.81	0.66

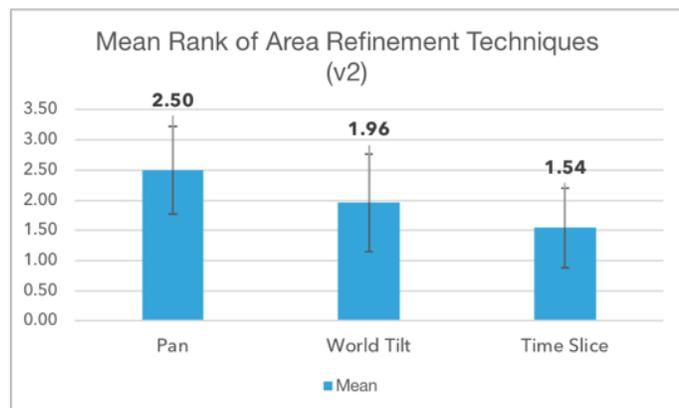


Figure 4.11: V2: Ranks assigned in order of preference to interaction techniques used to refine an Area of Interest. Lower scores are better (rank 1 is the best, while rank =3 is the poorest). Number of rank = 1 for each technique are: Time Slice (13), World Tilt (8), Pan + Zoom (3).

of, and expand upon how and why they ranked the various interaction techniques the way they did.

Participant feedback and discussion

In V1, World Tilt had performed particularly well. The same is true in this set, Time Slice and World Tilt both seemed to edge out Pan + Zoom in terms of user experience (Figure 4.11).

Time Slice: Before we discuss performance and user feedback, I would like to point out a characteristic unique to time slicing. Unlike Pan + Zoom, or World Tilt, Time Slice does not afford vertical / height-wise movement; participants could retrace their steps, but the eye (camera) remained at the car's window level (Figure 4.6). Therefore, if participants wished to reveal a hidden

or partially occluded target for easier selection after using Time Slice, they had to use an additional technique.

This played out in two ways during the test sessions: participants either continued to retrace their steps using Time Slice till they found a target that they could select easily, or, they enabled Pan + Zoom after Time Slicing and used it to refine the area of interest before selecting the target.

Of the total of 144 target selections with Time Slice (2 techniques to select an AoI * 3 target selections per technique * 24 participants), Pan + Zoom was used in 43 of them (~30%). More interestingly, the logs reveal that 7 of the 144 target selections were the smaller, hidden targets, and for all of them participants used Time Slice followed by Pan + Zoom. That is, the participants caught a glimpse of a smaller target while they were retracing their steps using Time Slice, but it was not easy to select the target without refining the scene further by using Pan + Zoom. While both Time Slice and Pan + Zoom turned out to be very useful by themselves, the two worked together to give very satisfactory outcomes.

Time Slice was ranked 1st among the three area refinement techniques by more than half of the participants (13 out of the 24 participants, i.e. 54%). Without any prompting, participants caught on to the utility of Time Slice. Here are a few excerpts from the questionnaire (the total number of participants in V2 made it impossible to embed it inline, it is recorded in Appendix F):

P6: “Time slice seems very useful, especially if something passed by unexpectedly.”

P8: “I found that Time Slice was more useful and effective at isolating my target in all situations than Pan was.”

P9: “Time Slice was very useful too in the faster moving car.”

P15: “Time slice is very useful in a real car situation when you are driving past things.”

P23: “Time Slice is neat. When I was young, on multiple occasions I wanted my sister to see something out of my side of the window but by the time I got her attention, it was too late. This would have been sooooo useful.”

World Tilt: The results were (encouragingly) quite similar to those of V1. From a total of 144 target selections using World Tilt, only 11 (~8%) needed to make use of the pan, zoom or rotate operations. Eight of the twenty four participants rated World Tilt as the best technique for refining the AoI, next to the thirteen for Time Slice (Figure 4.11). Again, some participants did end up selecting the smaller, partially occluded targets (16 times) exclusively, without using Pan + Zoom. Some excerpts from the questionnaire (Appendix F:) about World Tilt in V2 are as follows:

P6: “Being able to shift perspective is the most helpful, as well as engaging to me to get a better look at something going by”.

P8: “World Tilt was very useful for target far away or obscured by other objects.”

P9: “The world tilt allowed me to see targets clear (*sic*) and put them in context of their 360-degree surroundings. “

P11: “World tilt was also really cool since it allows you to see from an otherwise inaccessible (*sic*) perspective.”

P15: “World tilt is useful because it allows you to look around the area you are in.”

Having a bigger sample of participants revealed that there was some disagreement about the direction of the two-fingered vertical movement for World Tilt. Despite the initial training runs, a few participants (6 to be precise) fell into the habit of trying to move their fingers in the opposite direction from the one required (so instead of moving towards a bird’s eye view, they kept hitting the road surface). During the debriefing afterwards, all of these participants that had trouble with the directionality of World Tilt mentioned that they were trying to replicate the scroll direction on

their computers in AutoPoint. Just having an option to change the direction of the movement for World Tilt would have taken care of this issue.

Choice of Interaction Technique: To select an AoI, participants exclusively used the technique they preferred, throughout all target selections. Of the 24 participants, 19 used touch to take a snapshot, while 5 used point + gesture exclusively for the AoI selection phase.

From the selected AoIs, a total of 360 targets were selected (3 durations x 5 targets per duration x 24 participants). In 122 instances, the participants opted for a refinement process --- World Tilt was used 58 times, Time Slice 31 times and Pan + Zoom was used 33 times. In the remaining cases (238), the participants just tapped the target right after selecting an AoI, without attempting any kind of refinement. Of these 238, 21 target selections involved the easily obscured smaller targets.

With the duration parameter (the proxy for a car's speed), I did not find any notable variation in the frequency of use of Time Slice. I have a theory for this: by the time the participants started with the less structured condition (i.e. they could choose whatever interaction technique they preferred), they were familiar with the targets and their locations around the 3D world. Participants could almost predict when and where the next target would appear (or in the case of the smaller targets, have a landmark that obscured it), and take a snapshot very promptly. This theory seems to be borne out by the high percentage of target selections that did not use any area refinement technique (238 / 360, i.e. 66%), and also leads to the conclusion that the area refinement techniques would be the most useful while traveling through unfamiliar locations.

I was also curious as to why World Tilt was used more often than Time Slice, given that more participants ranked Time Slice better than World Tilt. Trying to correlate the log data with the questionnaire data for the participants revealed that there were four participants who had ranked Time Slice higher than World Tilt, but used World Tilt more. I do not have the data to explain this curious

anomaly, but again, I have a theory: either aesthetically, or intellectually, they liked Time Slice more but, when selecting the target, the larger area afforded by World Tilt proved to be more convenient for them.

Excerpts from Audio Recordings: After a test session was over, I asked participants to answer a couple of questions, and audio recorded their response. Though the questionnaire already asked participants to rank the three interaction techniques (Pan + Zoom, World Tilt, and Time Slice) they used to refine an area of interest, I wanted them to elaborate on their rationale for the rankings. I have cited just a couple of those responses pertaining to World Tilt and Time Slice, below:

Q: Among the three interaction techniques, Pan + Zoom, World Tilt and Time Slice, which of them did you particularly like or dislike, and why?

On World Tilt:

P16: “I think the most compelling on its own merit is the perspective tilt, partly because it allows a different perspective from something I am not going to get from the car at all. It allows me to say, hey what was that, then engage with Augmented Reality type representations / renditions of something architectural, or something like that.”

P19: “I like the tilt one I think, because it moves whatever is blocking what you want to see away from blocking it (*sic*).”

P8: “I ranked World Tilt and Time Slice as the most effective, I thought the World Tilt was really useful for things in the background, often the mountains were, you know, quite far away, and tilting would help select them. The same went for smaller buildings obscured by others, and you can see things that you can’t actually see from your window. That’ll be really cool because somewhere like Chicago if you drag up, you can see so much more of what’s on the street. (*sic*)”

On Time Slice:

P7: “I thought Time Slice was a better version of pan. I get that it’s not the same kind of pan, one is like the snapshot and the other is like going back in time, and I thought that Time Slice in almost all situations was more useful ... for like getting to the building I like.”

P20: “Time Slice for sure, because you already saw ... like ... what’s gonna be there that you missed something. Panning is definitely useful, you wanna explore the area, but I think it’s more of like a pastime, like let’s see what’s around here vs. oh shoot, I need to find that; time slicing allows you to do that.”

P3: “I really like the ... is it called time span? ... I think it would be very useful to me ‘cause sometimes I don’t realize that I wanted to see something until after it goes by, I’ve been in situations like that before when I wanted to backtrack, and this seems like a great way to do it. Specially because this is out of a passenger seat, you are facing to the side, so in order to see something that you passed, you have to turn your head around and that’s really hard, so I like this feature (*sic*)”

Overall: With this final iteration of AutoPoint’s development and testing in the lab, I have reported on the limitations and successes of various multi-touch and in-air gestural inputs when a 2D interactive area is all that is available to observe or learn about an external 3D world in motion.

Pointing clearly did not perform as well as I had anticipated. An overwhelming number of participants (30 out of 36, across Phase-I and Phase-II) preferred touch input to select an Area of Interest. The participants who did prefer pointing, mentioned that the technique was more fun to use, and I think the novelty of the interaction informed that opinion. However, I think there is a case to be made for a multimodal system that supports both pointing and touch; as P15 noted “...someone who’s on the middle seat or other side of the car can also do it without having to reach too far”.

AutoPoint’s final set of interaction techniques for engaging with the external world include the basic multitouch gestures (pan and zoom), a gesture to change perspective (World Tilt), and a feature that could help revisit an element that you missed (Time Slice). This covers most of the scenarios that one might find oneself in while traveling in an automobile, or for that matter, just

about any form of transport. Many participants referred to tours and education as possible applications for this kind of technology, but I like P14' s take on this the most: “it is almost as like you have a museum in your car”.

CHAPTER 5

AutoPoint: Phase III

For the third and final phase of AutoPoint’s development, I took the system and the features that worked well in the lab out into the real world. I fitted a car with an iPad that replicated the lab hardware setup and software capabilities as closely as possible. I drove 14 participants on a loop around campus to test the system in the real world and get their feedback.

5.1 Hardware: In-car iPad mount and recorder setup

The display assembly from the in-lab setup would have been nearly impossible to retrofit in a car (without even thinking about using a projector based system in a well lit environment). A good alternative was to find the biggest tablet with a back facing camera and mount it on a car window.

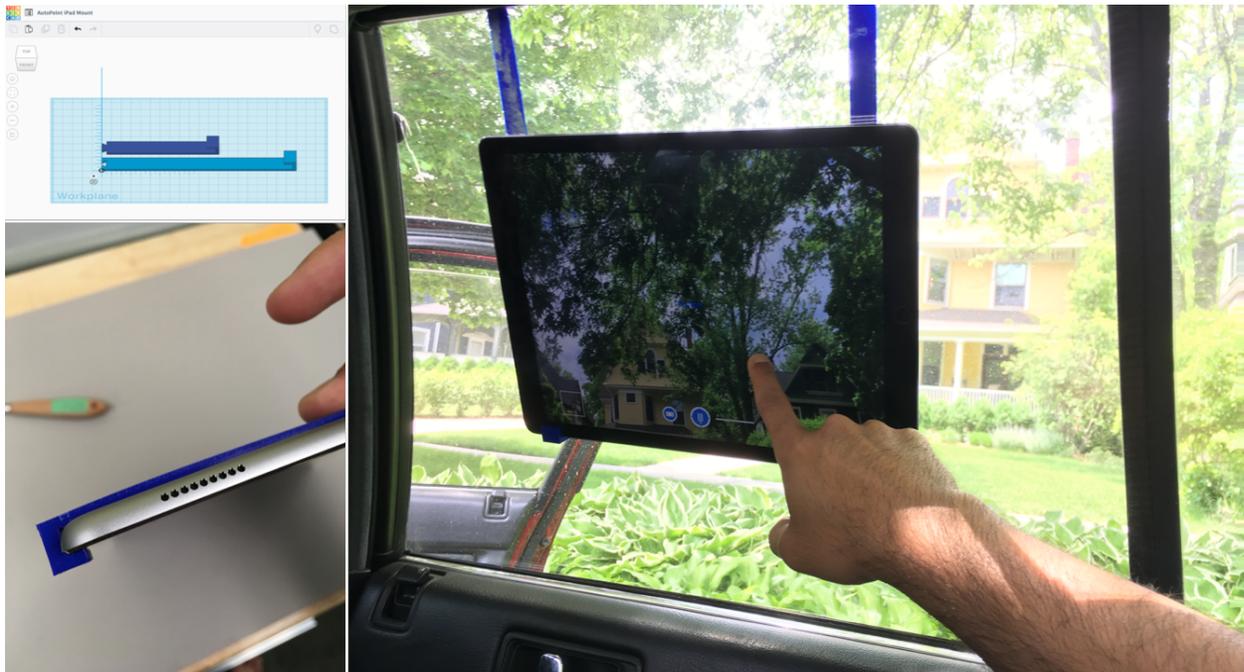


Figure 5.1: Initial design of the mount was an elongated s-shaped hook that supported the iPad on one end and hooked onto the top of the window on the other. Velcro strips on the back of the iPad kept it flush with the window.



Figure 5.2: Blacked out rear window (top-left), a platform with a lip that the iPad rested against (top-right), and the iPad's camera aligned with a hole in the blacked out window (bottom).

I chose a 12.9" iPad Pro for my application. The idea was to show the video feed from the iPad's camera on the screen continuously. If the iPad was mounted judiciously, the iPad and the image from its back camera could serve as a reasonable approximation of a transparent display.

I needed two iterations to get a workable mount design. During a test with the first iteration (Figure 5.1) of the mount, I realized that even though its design made the iPad very secure, it did not work well with a slight curve in the window. The window's curve resulted in the camera being pointed too high, so that the image on screen did not look realistic. In addition, this test session also indicated that I had to black out the window. The backlight from the surrounding clear glass made it difficult to view the iPad screen. Also, the discontinuity in the field of vision, where the



Figure 5.3: iPhone used to video record the test session and provide cellular data to the iPad.

views through the iPad camera and the surrounding clear glass did not blend together, proved to be a bother for the user (Figure 5.1).

In the next iteration, I blacked out the car window with chalkboard paper [61]. I fashioned a mount from scrap steel and screwed it to the top of the door panel (Figure 5.2, top tight and bottom). The bottom of the iPad rested on the lip on the mount, while the window and the top of the iPad had appropriately placed Velcro strips to make this assembly secure and prevent the iPad from tipping over. There was a small hole cut into the chalkboard paper that lined up with the iPad's camera. I ran a power cord to the iPad as well, to ensure that it did not run out of charge during a test session.

I used an iPhone mounted on the driver's headrest to video record sessions of participants using the system. This iPhone did double duty as the cellular data provider for the iPad application (which needed data for a map view).

5.2 Software: iPad Application

AutoPoint’s iPad application was written in the Swift programming language [62]. Two of Apple’s iOS frameworks, namely, MapKit and AVFoundation, were used extensively to help build it. The iPad application had two distinct views: Live View, and Map View. The application also allowed users to save and revisit elements of interest that they came across, or discovered during a drive, by taking photos, saving information from the Map View and so on.

5.2.1 Live View

In this view the iPad showed a live camera feed on the screen. In terms of features, a user could take a photo, enable Time Slice, Interact with Times Slices and look at the snapshots – photos and Time Slices – they collected during a drive (Figure 5.4).

Take Photo: This worked just as any regular camera application would. A user could tap the camera icon (Figure 5.4, left) to capture a snapshot of the scene currently displayed on the screen. Initially, a smaller preview of the captured imaged would be shown on the bottom left of the screen. The preview did not obscure the live video feed in a major way. If satisfied with the preview, the user

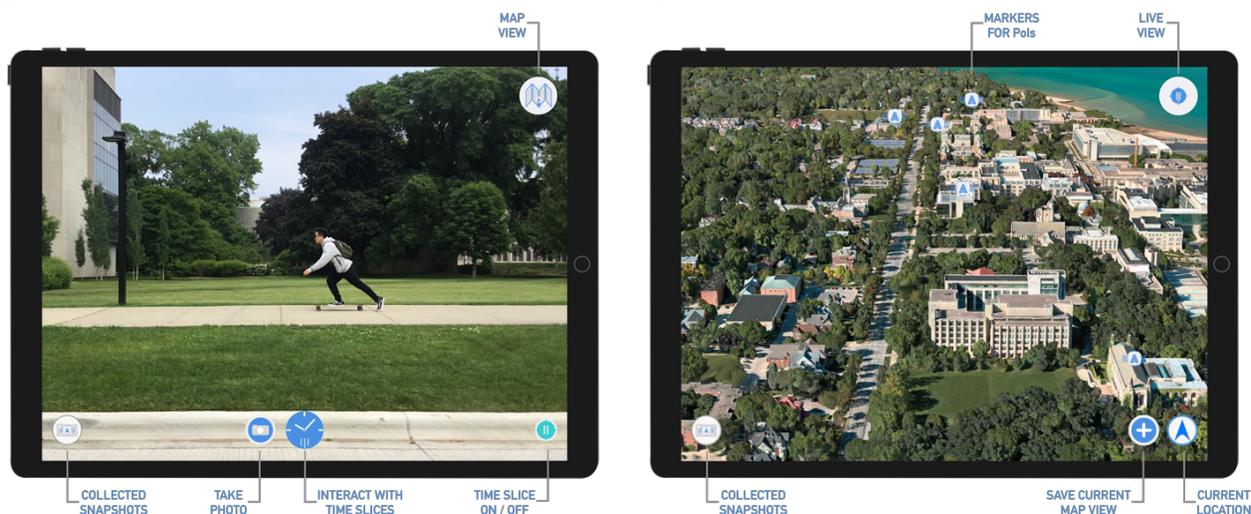


Figure 5.4: Two distinct views, Live View (left), and Map View (right). Live View allows Time Slicing, taking a snapshot, and reviewing saved snapshots. Map View allows one to look at POIs, explore by using pan and world-tilt gestures, and also review saved snaps.

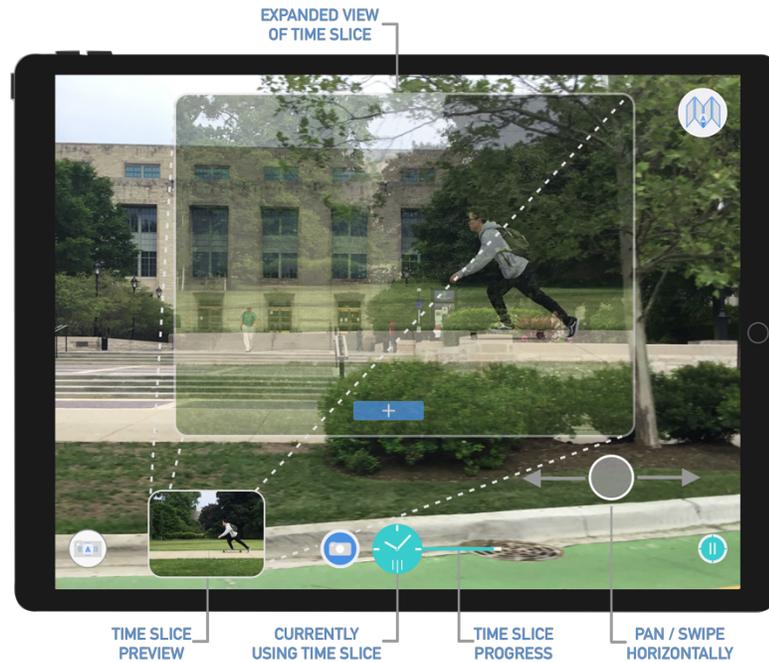


Figure 5.5: Time Slice initially shown as a small preview after pressing the "clock" icon. Panning/Swiping with one finger allows one to go back and forth in Time Slices. Tapping the preview makes it bigger. There also is a progress bar showing current position of Time Slices.

could tap a “plus” (add) icon to save this image. I expected this feature to be more useful when the car was stationary, e.g. at a stoplight.

Interact with Time Slice: The participant engages with Time Slices by tapping on the "clock" icon (Figure 5.4, blue icon at bottom center of the screen). As with the photo capture feature, the particular time slice is first seen as a small preview at the bottom left of the screen; tapping it makes it larger. Swiping with a single finger takes one back and forth in time.

Compared to the in-lab setup, there are a couple of obvious changes to the visualization for this version: a) in both the minimized and expanded views, a portion of the real world continues to be visible in the background, and b) a progress bar serves as an indicator of the remaining time slices that participants could traverse. Both these features were introduced based on the feedback and observation of participants in the lab. Keeping the real world visible in the background



Figure 5.6: Map View showed a satellite view initially centered on the current location. A two-fingered pan/swipe (same as World Tilt in the lab) showed the 3D representation. PoI's were annotated with "A" markers. A user could save the map view or an image of the PoI.

reduced the chances of participants feeling disoriented after they stopped using Time Slice, when the car would have moved to a new location, thus abruptly presenting a very different scene.

The Time Slice feature was implemented using a circular buffer very similar to the in-lab setup. Only this time, instead of storing the camera's position and orientation, I stored an image taken by the camera every 100ms (i.e. ten snapshots per second). The size of the buffer was 120 frames, thus giving a user twelve seconds worth of images to use as Time Slices.

5.2.2 Map View

To go to the Map View (Live View was the default View), a user tapped on a map icon on the top right (Figure 5.4), or anywhere around the middle of the screen. In this view the iPad showed an interactive satellite view of a map. The map started out being centered on a user's current location. GPS coordinates were received from the iPhone that was doing the video recording during the test sessions, with a circular blue icon showing that current location (Figure 5.6).

In terms of features, a user could tilt the 2D map to show elements in 3D (World Tilt). Tilting the map also revealed interactive PoIs (Figure 5.6). A user could touch the annotation for a PoI to get historical data (including photos) about that PoI. I sourced most of these historical data from Northwestern University's Deering Library Archives. A user could also Pan + Zoom in this Map View, and save an image of the map currently displayed on the screen.

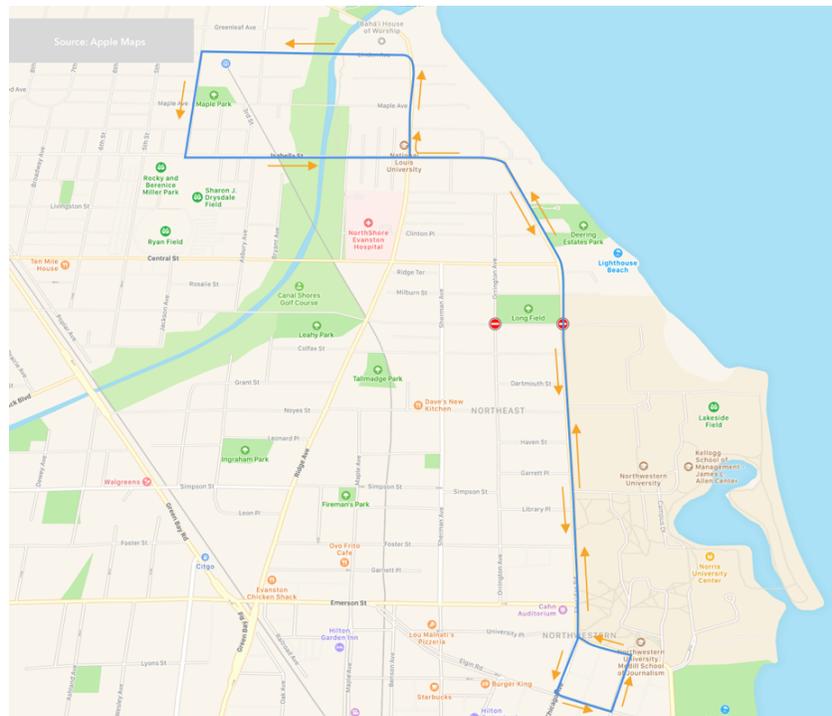


Figure 5.7: The loop driven around and north of campus.

5.3 Test Sessions

5.3.1 *Participants and testing parameters*

To test this version of AutoPoint, I drove fourteen participants (8 Female and 6 male, between 22 to 31 years of age) in a loop around Evanston (Figure 5.7). Of these fourteen participants, eight had participated in the second iteration of AutoPoint (i.e. Phase – II).

Prior to the drive, I met the participants in the lab and demonstrated the various features of the AutoPoint iPad application. I asked the participants to put the application through its paces in the lab to reduce the probability of having to intervene during the drive. After the participants reported being comfortable with the various interaction techniques and features, they were taken to the car parked nearby and the test session commenced.

During each session, two participants occupied the back seat of the car. Half way through the route, they traded places. The participants were instructed to simply use the system and think aloud about what they were trying to do. They were free to interact with each other (e.g. to discuss something of interest along the route) at any point of time.

At the end of a test session, the participants were debriefed about their experience of using the system. This part was open ended (and relatively unstructured), but all the participants were asked to share their overall impressions, which features stood out for them, what worked or did not work, and to compare and contrast the two basic modes of operation (Live View and Map View). I video recorded the entirety of the in-car session, which included the debriefing (using the setup shown in Figure 5.3). The overarching goal was to capture user interaction patterns that could reveal design shortcomings and areas for improvement.

5.3.2 Participant feedback and discussion

Some interesting interaction patterns emerged from the testing sessions. Those patterns along with participant feedback about the various interaction techniques and gestures are outlined below (*in the reports that follow, quotes from participants are reproduced verbatim, errors and all*).

Stabilizing the hand: while the car was in motion, many participants devised a way to stabilize their arm/finger while interacting with the system. The stabilization technique differed from person to person. Examples include using the iPad mount (Figure 5.8, left), or using a combination of the door frame to rest the forearm and the iPad's bezels to stabilize the wrist (Figure 5.8, middle), or even using one arm to support the other forearm (Figure 5.8, right).

The frequency and duration of using a stabilization technique depended on factors like vehicle speed, road surface quality, and the task a participant was trying to accomplish. For example, since the Time Slice feature required the participants to swipe through contiguous frames of data before selecting an image, many participants stabilized their hand. On the other hand, while taking a photo – regardless of whether the car was stationary or not – participants just tapped the camera button (Figure 5.4, left) and chose to forego any sort of stabilization.

Similarly, slightly more intricate gestures like the two-fingered World Tilt in the Map View required participants to maintain contact with the touchscreen. If one of the two fingers lost contact



Figure 5.8: Many participants devised a way to stabilize the hand before hitting a target.

with the touchscreen, the gesture recognizer failed, and participants had to repeat that gesture. In this case too, I noticed that the participants tended to use the iPad's bezel to stabilize their wrist.

These observations from the video data are also supported by participant feedback during the post drive debriefing. For example, *PI* said “I would put my thumb on something (*pointing at the iPad's bezel*) ... because then I am vibrating with the car and with the iPad”.

Observing the system in use in the real world (also borne out by participant feedback) seems to suggest that any touch-based interaction techniques that require a modicum of finesse or accuracy need to account for coarser inputs. In AutoPoint's case, both the two-fingered World Tilt and the Time Slice interactions would have benefitted from having better support for coarser inputs.

An example of supporting coarser inputs would be to implement gesture recognizers that are more forgiving. For the World-Tilt gesture (Figure 5.6), a time buffer could have prevented the two-fingered swipe from being invalidated when one finger momentarily lost contact with the window (i.e. the iPad screen). This would have resulted in fewer failed attempts. These gesture recognizers can be designed to be adaptive to the car's state of motion. When the car is stationary, accidentally losing a point of contact while swiping is unlikely (hence no time buffer is required). The opposite would be the case when the car is in motion; one can also imagine a scenario where the length of this time buffer can be made variable depending on the road surface quality.

Another way to tackle this issue can be to incorporate physical controllers (e.g. a joystick) onto the door pad. That being said, while a door pad mounted input device might help with fine-grained control on bumpy surfaces, it has some obvious limitations (similar to most indirect input modalities). Decoupling the display and the input for something like the World Tilt interaction would make the overall experience less fluid, with more frequent mode switching (take, for example, when a passenger wants to use World Tilt followed by a pan gesture). On the other hand,

since Time Slicing is a stand-alone interaction – you either go backwards or forwards in time – with less potential for mode switching, a physical control could theoretically be advantageous. This design tension is quite similar to car dashboards where we see a trade-off between functions that are offloaded to a touchscreen display versus ones that continue to have a manual override.

Overall, in the real world, where road conditions affect input accuracy tremendously, support for coarse inputs is critical.

Live View and Map View: In the Live View, participants were able to skim through 120 frames (one frame every 100ms) of image data using Time Slicing. All the fourteen participants – including the six participants who had not participated in the lab study – noted that being able to revisit an instance in time was appealing. Here are a few of excerpts from their feedback:

P1: “Snapshot functionality was really good. Especially when you are in the spur of the moment, you are trying to talk but you have seen something interesting ... but I just missed it (*sic*)! Snapshot allows you to get back and get that picture.”

P2: “It can be anything from a tourist activity to just a simple oh, that's an interesting looking tree that I might want that in my landscaping, or something random like that, you might just need that picture ... and it gives you a moment to pause.”

P5: “If I were to see people coming by and saw something interesting and saw someone wearing something interesting, I'd want to go back and pause and collect it for later ... shopping.”

P8: “Check this out (*drawing the other participant's attention*). All the time you are driving, and you like, miss the one spot where you could have taken a good picture because you have stuff in the way? Well, you can time slice and then go back to get it just right.”

Based on my observations of the test sessions, there are some enhancements that would improve the experience of using Time Slices. For example, with the current implementation, if a vehicle is

moving fast, the Time Slices might not be granular enough to sharply capture an object of interest. Making the interval between image frames variable and dependent on vehicle speed would solve that issue. Similarly, the sensitivity of the gesture itself – i.e. the horizontal pan – can be made variable to account for cases where a participant wants to select a precise image frame. Two participants mentioned this:

P2: “I do think the sensitivity is a bit fast if there are bumps on the road. A little bit of the sensitivity, like a 2x, 4x speed change in the buffer would have helped.”

P8: “The scroll, maybe, some kind of an adaptive scroll so that you could have precision to go frame by frame but then have it scroll faster if you were just going side to side might help.”

In the lab setup I had the luxury of designing the 3D world and had complete control of the camera’s movement (to simulate the view from a moving car). On the other hand, in this version, I did not have a rich (real time) 3D-model of the space we were going to drive through. In order to have feature parity with the in-lab implementation of AutoPoint, I had to design the iPad application to have two distinct modes of operation: Map View to interact with PoIs and Live View with Time Slicing. In the post drive debriefing, I had asked participants to express their thoughts on these two modes, and to let me know if they had a preference between the two:

P1: “... in particular using the map for more of when you are not entirely sure of what to be looking at, you can use the map as a guide of things to suggest, particularly when you are at a stoplight, or kind of like in a blank space, you can use that map to look at features that would be not the ones you missed, but also the ones that you expect to come along. So not just history, but sort of preemptive and getting ready.”

P8: “I enjoyed being able to switch between it. For me, the map view gave me a sense of the bigger picture of where I’m going, like GPS feature, it almost allowed me to get a sense of where we were and anticipate what might be coming up later.”

P5: “I typically like the live view. It made me feel much more observant of what I was seeing. But I also liked that the map view because I’m particularly interested in history, it provided facts and background that you wouldn’t know otherwise.”

The strategy of using the Map View to plan ahead is interesting to me, especially because this was not a conscious design decision. But being able to freely explore the world nearby without being constrained to the slice of the world visible from the car window seems like an important design factor, especially if we were to consider a purely AR implementation in the future.

That said, there are trade-offs with this design approach. Spending time in the Map View could lead to users missing out on something interesting or feeling slightly disoriented upon their return to the Live View. This thought was echoed by some participants:

P1: “I didn’t like that you have to go in map mode and you can’t have that window any more, like it’s one or the other. I kind of wanted a split screen or at least a sub-quarter of it.”

P3: “*(while in map view)* I was wondering if I missed out on something. in the meantime. It’s nice, but because you can do so much with it, I feel like I might be missing out on what is going on in the real world.”

P5: “*(while in map view)* It was really interesting, but I feel like that took me more out of the present moment. When I got really involved in that, then 5 or 10 minutes down the road I would have no idea what was out the window.”

P12: "I think I got sucked in to the map mode more, because I’m really fascinated with how things are laid out and how they look from the aerial view."

I expect this design constraint to be alleviated when we have more accurate real time location tracking and a more detailed 3D model of the world. The Live View would incorporate 3D/depth data, and similar to the setup in the lab, a user would be able to explore the world without a loss of context arising from mode switching.

Potential as an Informal Learning Environment: With two people in the back seat of the car for each test session, I was curious to see what kinds of conversations the participants engaged in, and whether those conversations could be directly attributed to using the AutoPoint system. More specifically, I was looking for instances that could highlight AutoPoint’s potential as a place-centered informal learning environment [5]. Upon reviewing the video data, I found many instances (all test sessions had at least one such occurrence) that were encouraging. For example, one participant discovered a new landmark and drew the co-passenger into a conversation about that landmark:

P7: “We have a lighthouse!?”

P8: “Yeah.”

P7: “I had no idea. How big is it?”

P7: “It’s pretty big. Let me see if I can bring it back up in the time lapse ...”

P8: “Why do we need a lighthouse here?”

By switching to the Map View, they were able to pull up information about the lighthouse (it was highlighted as a PoI). P7 and P8 proceeded to speculate why the area we were driving through needed one. On the way back (we were driving in a loop, Figure 5.7), P7 looked up the lighthouse again and even saved a photo of it as a part of her collection.

While the above is an example of a PoI discovered along the test loop, there also are instances of conversations revolving around PoIs that both participants were already familiar with (i.e.

AutoPoint triggered a conversation around this familiar object). For example, coming upon an area close to the University Main Library, P9 and P10 discussed its architecture:

P9: “That's a pretty interesting photo of the Main (*Library*).”

P10: “I feel like Brutalist college architecture does look good ... sometimes.”

P9: “Especially in black and white.”

P10: “I think what they should do is acid sand it so that it is sparkly white again, or even paint over it.”

P9: “Right, they should paint over it. It does look cool. Architecturally, this is very intriguing”.

Note, this building that the participants were talking about was not visible from the car. P10 was exploring the area in the Map View and used World Tilt to show the library building in 3D. Thereafter, P10 drew P9 into the conversation about that building’s architecture. I also found other instances of participants engaging in a more collaborative exploration of the space we were driving through. For example, P13 and P14 were both interested in architecture and consulted each other before saving photos of homes that they encountered during the test session.

While I cannot quantify AutoPoint’s role in triggering or satiating curiosity, I think it is fair to say that the test sessions revealed a variety of ways in which AutoPoint structured and supported engagement with the world the participants were travelling through.

Other Feedback and Observations: While the utility of Time Slice was apparent to all participants, some of them brought up the privacy implications of this feature. Since privacy is a complex concept and its interpretation varies from person to person, I received fairly wide-ranging feedback about it. Here are a few relevant quotes from the participants:

P7: “as a woman walking on the street, I would be a little unnerved about people’s opportunities to take photos of me and save them forever”

P11: "I think that anyone could just hold up their iPhone up to the window and record, so why can't this also be a thing? It allows you to be more sneaky about it. So that is a little bit concerning. "

P12: "I'm not concerned about privacy at all."

P13: "I think there are a lot of things that already track you. But I'm glad that the time slicing doesn't go forever."

Even though smartphones – with their rapidly improving cameras – already make it fairly easy for a malicious actor to invade someone's privacy, I think it is important to think through AutoPoint's implementation details that could bolster privacy. For instance, the images a user can Time Slice through could have people's faces (picked up by the camera) blurred out by default. Even something as simple as having an indicator (similar to an LED that video cameras used to have), to warn passerby that they might be on camera, seems like the right thing to do. That said, the current hardware implementation of AutoPoint with the blacked-out window (that hid the iPad), might have made it seem more surreptitious than a system with a transparent display/window in its place.

Two of the fourteen participants reported feeling slight motion sickness while using AutoPoint over rougher sections of the road. Both of them reported that just averting their gaze from the iPad for a quick second and facing the front of the car diminished the feeling rapidly. In a non-test environment where users would tend to interact with the system in short chunks of time, motion sickness should not be a major issue. It must be said that the significance of real world testing is underscored by the fact that neither privacy, nor motion sickness, cropped up as potential issues in any of the in-lab test sessions.

CHAPTER 6

Conclusion

With AutoPoint, I explore a problem domain that I refer to as MTDI-EWM (Multimodal Transparent Display Interactions with an External World in Motion). More specifically, which multi-touch and pointing gestures can be used to engage with an external world in motion using an intermediary transparent display. I focused on automotive touring, and how passengers could forge richer connections with the areas they were driving through, and whether the automobile itself could scaffold such interactions.

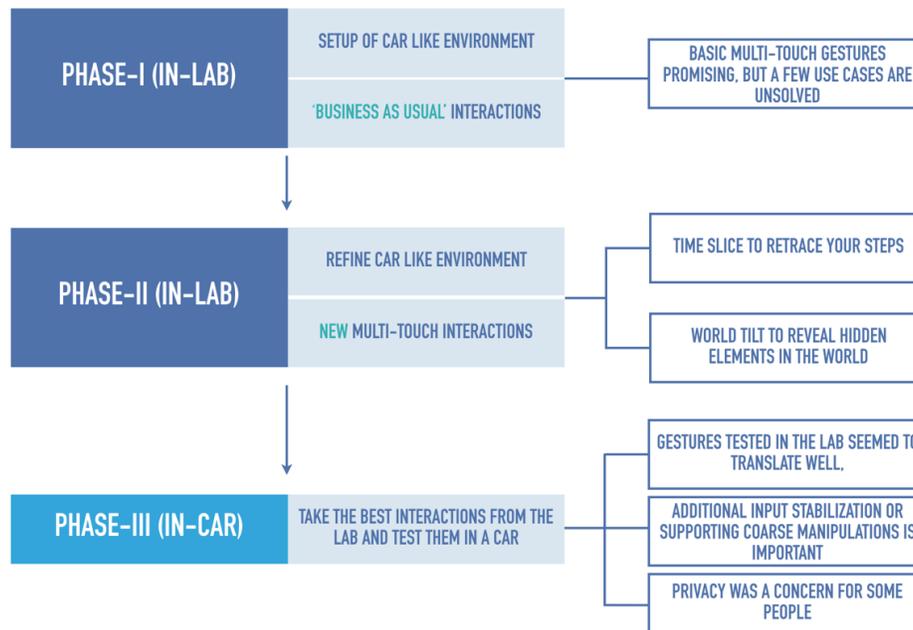


Figure 6.1: An overview of the various phases and outcomes from AutoPoint's development.

6.1 Problem Domain and Iterative Development

The overarching problem domain can be characterized as follows:

- Temporal: how can a user interact with a PoI that just passed by (either because the car was traveling fast, the object was too close to the car, or the user was not paying attention)?

- Geo-Spatial: how can we determine just what item a user is targeting? It is non-trivial to figure out which PoI a user is pointing at; it could be very close, miles away, or somewhere in between along the line of pointing.
- Location or Map based: how can we isolate and identify a particular PoI in a densely packed space?
- Social in Nature: how can we support co-located users? Automobiles can be shared spaces with social interactions (e.g. drawing a co-passenger's attention to something seen outside the window).
- Informal learning: can an application built around the factors already mentioned transform a car ride into a place-centered informal learning environment [5]?

Based on the framework outlined above, I developed AutoPoint in three phases. The first two phases were in the lab, while the third, and final, phase of development was done in the real world (Figure 6.1).

Setting up a car-like environment in the lab was a major component of Phase-I. Along with the hardware setup, I investigated how well basic multi-touch (pan, zoom, and rotate) techniques work for this problem area and explored their strengths and weaknesses. The target selection process was split into the following sequence of occurrences: 1) from the external world outside the car window, select an area of interest; 2) refine that area of interest; 3) select the target; and 4) return to view the external world.

In Phase-II, the car-like environment in the lab was refined, and new multi-touch interactions were developed. The research questions were: 1) how can multi-touch gestures be used to reveal an element that is obscured from a user's view? and 2) how could we solve the problem of missing

out on something interesting because it went by too fast? A total of 31 participants used the system in the lab and provided feedback about the interaction techniques. The World Tilt and Time Slicing gestures seemed to be highly effective in addressing the research questions. In this phase, participants tried using a pointing gesture to select an initial area of interest. However, they preferred tapping the window (i.e. touch input) over the pointing gesture.

The goal for the final iteration of AutoPoint was to take the best interaction techniques and features from the lab, and test them in the real world. Using an iPad as a proxy for a transparent display, this round of testing involved fourteen participants. This real world testing revealed a couple of facets that had not come up in earlier rounds of development: 1) the need to have in-built stabilization or support for coarse inputs leading to finer interactions (e.g. using Time Slice to select a single image among many), especially on bad roads, and 2) the privacy implications of such a system. Test sessions also confirmed that using AutoPoint elicited curiosity and engagement with the external world, thus showcasing AutoPoint's potential in harnessing automobiles as informal learning environments.

6.2 Future Work

Overall, the fundamental premise of AutoPoint – to engage automobile passengers with the external world – was shown to be quite promising. Building upon this research, I am interested in exploring how artificial intelligence, machine learning, and computer vision based techniques can be applied to improve overall usability. For example, one of the techniques that I discarded in Phase-II, namely the depth-of-field visualization, could benefit tremendously from computer vision based enhancements, whereby, given an accurate 3D model of the space, the system could be much smarter in recognizing (and highlighting) potential PoIs.

Similarly, for the class of problems (MTDI-EWM) that I explored with AutoPoint, potential techniques include multi-touch input, gestural input, AR, and speech-based or physical controls. Of these four, I only explored multi-touch input in depth, and gestural input somewhat cursorily. While gestural input did not seem to perform as well as touch input in my testing, it is not clear that participants would reject gestural input given a highly accurate gesture detection system/hardware, especially in cases where touch input is not feasible (e.g. for a passenger seated in the middle). Then there is the option of supporting multimodal input. For example, natural language input can be combined with physical controls (like a button on the armrest) to select PoIs. And those are all options at the input level; further options become available at the higher levels.

Apart from the purely technical challenges, there are many other potential avenues of research. Within the broader context of MTDI or MTDI-EWM, there are several use cases worth exploring. Scenic overlooks equipped with binoculars can be re-engineered to use transparent displays with support for multi-modal input (MTDI). While scenic overlooks generally have static features (e.g. valleys, mountains, city skylines etc.), a transparent display affords interactions — like World Tilt — that could reveal hidden or obscured features. One can also imagine interacting with these transparent displays to transform the scenery to simulate various seasons and weather conditions (the interaction could be quite similar to Time Slicing, albeit on a different scale).

Public spaces like zoos and aquariums could be fitted with transparent displays that allow visitors to view and record animal behavior and support interactive educational experiences (e.g. selecting a particular fish within a school and learning more about it). If these interactive experiences are designed judiciously, transparent displays can strike a balance between visitor preference for display-free or display-rich environments. Within the vehicular context — as with AutoPoint — there are unexplored use cases. For example, since cars are shared spaces, what are

the ways a system can support collaborative use? Similarly, if we extended this class of technology to a larger scale, like a school bus with many transparent displays, could it enhance the daily commute or the occasional field trip? What would turn taking look like in a multi-user setting, or how can information be shared between passengers, each having a transparent display by their side?

References

1. Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2011. Pointable : An In-Air Pointing Technique to Manipulate Out-of-Reach Targets on Tabletops. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*: 11–20. <https://doi.org/10.1145/2076354.2076357>
2. John Bichard, Liselott Brunnberg, Marco Combetto, Anton Gustafsson, and Oskar Juhlin. 2006. Backseat Playgrounds: Pervasive Storytelling in Vast Location Based Games. . 117–122. https://doi.org/10.1007/11872320_14
3. Nora Broy, Sebastian Goebel, Matheus Hauder, Thomas Kothmayr, Michael Kugler, Florian Reinhart, Martin Salfer, Kevin Schlieper, and Elisabeth André. 2011. A cooperative in-car game for heterogeneous players. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 167–176.
4. Xiang Cao and Ravin Balakrishnan. 2003. VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, 173–182.
5. National Research Council and others. 2009. *Learning science in informal environments: People, places, and pursuits*. National Academies Press.
6. Rao; Manoharprasad K. Cuddihy; Mark A. 2013. Ford self-driving car entertainment system. Retrieved July 13, 2018 from <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsearchbool.html&r=1&f=G&l=50&col=AND&d=PTXT&s1=9272708.PN.&OS=PN/9272708&RS=PN/9272708>

7. Rick DeMeis. Heads Up! Here Come Automotive HUDs. 2013. Retrieved July 13, 2018 from <https://electronics360.globalspec.com/article/3733/heads-up-here-come-automotive-huds>
8. Michael Gleicher and Andrew Witkin. 1992. Through-the-lens camera control. *ACM SIGGRAPH Computer Graphics* 26, 2: 331–340. <https://doi.org/10.1145/142920.134088>
9. Charles Goodwin. 2003. Pointing as situated practice. In *Pointing: Where Language, Culture, and Cognition Meet*. 217–241. <https://doi.org/10.4324/9781410607744>
10. Martin Hachet, Fabrice Declé, Sebastian Knödel, and Pascal Guitton. 2008. Navidget for easy 3D camera positioning from 2D inputs. *3DUI - IEEE Symposium on 3D User Interfaces 2008*: 83–89. <https://doi.org/10.1109/3DUI.2008.4476596>
11. Hajime Hata, Hideki Koike, and Yoichi Sato. 2016. Visual Guidance with Unnoticed Blur Effect. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '16*, 28–35. <https://doi.org/10.1145/2909132.2909254>
12. Liang Hiah, Tatiana Sidorenkova, and LP Romero. 2013. Engaging children in cars through a robot companion. *Proceedings of the 12th ...*: 384–387. Retrieved January 8, 2015 from <http://dl.acm.org/citation.cfm?id=2485815>
13. Ken Hinckley, Randy Pausch, John C Goble, and Neal F Kassell. 1994. A survey of design issues in spatial input. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, 213–222.
14. John A Jakle. 1981. Touring by automobile in 1932 The American West as stereotype. *Annals of Tourism Research* 8, 4: 534–549.
15. John F Kelley. 1985. *Cal-a natural language program developed with the oz paradigm: Implications for supercomputing systems*. IBM Thomas J. Watson Research Center.

16. H. Kim, C. M. Gilmore, A. Piqué, J. S. Horwitz, H. Mattoussi, H. Murata, Z. H. Kafafi, and D. B. Chrisey. 1999. Electrical, optical, and structural properties of indium–tin–oxide thin films for organic light-emitting devices. *Journal of Applied Physics* 86, 11: 6451–6461.
<https://doi.org/10.1063/1.371708>
17. David M Krum, Jens Faenger, Brian Lathrop, Jo Ann Sison, and Annie Lien. 2008. All roads lead to CHI: interaction in the automobile. *Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems* 2: 2387–2390.
<https://doi.org/10.1145/1358628.1358691>
18. Andrew L. Kun, Albrecht Schmidt, Anind Dey, and Susanne Boll. 2013. Automotive user interfaces and interactive applications in the car. *Personal and Ubiquitous Computing* 17, 5: 801–802. <https://doi.org/10.1007/s00779-012-0520-7>
19. Felix Lauber, Anna Follmann, and Andreas Butz. 2014. What You See Is What You Touch : Visualizing Touch Screen Interaction in the Head-Up Display. *Proceedings of the 2014 conference on Designing interactive systems*: 171–180.
<https://doi.org/10.1145/2598510.2598521>
20. Eric Laurier, Hayden Lorimer, Barry Brown, Owain Jones, Oskar Juhlin, Allyson Noble, Mark Perry, Daniele Pica, Philippe Sormani, Ignaz Strebel, Laurel Swan, Alex S. Taylor, Laura Watts, and Alexandra Weilenmann. 2008. Driving and ‘Passengering’: Notes on the Ordinary Organization of Car Travel. *Mobilities* 3, 1: 1–23.
<https://doi.org/10.1080/17450100701797273>
21. S K Lee, William Buxton, and K C Smith. 1985. A multi-touch three dimensional touch-sensitive tablet. In *Acm Sigchi Bulletin*, 21–25.
22. James R. Lewis. 1993. Multipoint scales: Mean and median differences and observed

- significance levels. *International Journal of Human-Computer Interaction* 5, 4: 383–392.
<https://doi.org/10.1080/10447319309526075>
23. Andreas Löcken, Shadan Sadeghian Borojeni, Heiko Müller, Thomas M Gable, Stefano Triberti, Cyriel Diels, Christiane Glatz, Ignacio Alvarez, Lewis Chuang, and Susanne Boll. 2017. Towards Adaptive Ambient In-Vehicle Displays and Interactions: Insights and Design Guidelines from the 2015 AutomotiveUI Dedicated Workshop. In *Automotive User Interfaces*. Springer, 325–348.
 24. Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. 1990. Rapid controlled movement through a virtual 3D workspace. *ACM SIGGRAPH Computer Graphics* 24, 4: 171–176. <https://doi.org/10.1145/97880.97898>
 25. Kohei Matsumura, Tadashi Sakamoto, Haruo Noma, and Yasuyuki Sumi. 2015. Stick'n conversation: stick in-car conversation into places using multi person finger pointing gestures. In *Adjunct Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 26–31.
 26. Alexander Meschtscherjakov, Nicole Perterer, Sandra Trösterer, Alina Krischkowsky, and Manfred Tscheligi. 2017. The Neglected Passenger—How Collaboration in the Car Fosters Driving Experience and Safety. In *Automotive User Interfaces*. Springer, 187–213.
 27. Alexander Meschtscherjakov, David Wilfinger, Nicole Gridling, KATja Neureiter, and Manfred Tscheligi. 2011. Capture the Car! Qualitative In-situ Methods to Grasp the Automotive Context. *Proceedings of the International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '11)*: 105–112.
<https://doi.org/10.1145/2381416.2381434>
 28. Natasa Milicic and Thomas Lindberg. 2008. Menu interaction in head-up displays. In

Human Factors and Ergonomic Society, Annual Meeting, Soesterberg, The Netherlands.

29. Ditte Mortensen. 2018. Natural User Interfaces – What are they and how do you design user interfaces that feel natural? | Interaction Design Foundation. Retrieved July 13, 2018 from <https://www.interaction-design.org/literature/article/natural-user-interfaces-what-are-they-and-how-do-you-design-user-interfaces-that-feel-natural>
30. Jim Motavalli. 2010. The Electronic Systems That Make Modern Cars Go (and Stop) - The New York Times. Retrieved July 13, 2018 from <https://www.nytimes.com/2010/02/05/technology/05electronics.html>
31. Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air pan-and-zoom on wall-sized displays. In *Proc. CHI*, 177–186. Retrieved June 21, 2011 from <http://hal.archives-ouvertes.fr/hal-00559171/>
32. Oblong Industries. <http://www.oblong.com/>.
33. Raphael Orlove. 2017. This Is What A Hands-Free Car Phone Looked Like In 1988. Retrieved July 13, 2018 from <https://jalopnik.com/this-is-what-a-hands-free-car-phone-looked-like-in-1988-1791511875>
34. JS Pierce, AS Forsberg, MJ Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. 1997. Image plane interaction techniques in 3D immersive environments. In *Proc. Interactive 3D graphics*, 39–43. Retrieved December 15, 2010 from <http://portal.acm.org/citation.cfm?id=253303>
35. Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. 2009. A screen-space formulation for 2D and 3D direct manipulation. *Symposium on User Interface Software and Technology*: 9. <https://doi.org/10.1145/1622176.1622190>
36. CA) Rober, Mark B.; (Sunnyvale. 2018. Augmented Virtual Display. *USPTO*. Retrieved

- July 13, 2018 from <http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnethtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220180089900%22.PGNR.&OS=DN/20180089900&RS=DN/20180089900>
37. Sonja Rümelin, Chadly Marouane, and Andreas Butz. 2013. Free-hand pointing for identification and interaction with distant objects. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 40–47.
 38. P. Sundström, A. Baumgartner, E. Beck, C. Döttlinger, M. Murer, I. Randelshofer, D. Wilfinger, A. Meschtscherjakov, and M. Tscheligi. 2014. Gaming to sit safe: The restricted body as an integral part of gameplay. *Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, DIS*: 715–724.
<https://doi.org/10.1145/2598510.2600882>
 39. Riccardo Toppan and Mario Chiesa. 2015. Integrating a Touchless UI in the Automotive Environment. *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI '15*: 1–4. Retrieved July 13, 2018 from <https://www.auto-ui.org/15/p/workshops/5/toppan.pdf>
 40. Daniel Vogel and Ravin Balakrishnan. 2005. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST*, 33. Retrieved from <http://portal.acm.org/citation.cfm?doid=1095034.1095041>
 41. Daniel Wigdor and Dennis Wixon. 2011. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier.
 42. David Wilfinger, Alexander Meschtscherjakov, Martin Murer, Sebastian Osswald, and Manfred Tscheligi. 2011. Are we there yet? a probing study to inform design for the rear

- seat of family cars. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6947 LNCS, PART 2: 657–674.
https://doi.org/10.1007/978-3-642-23771-3_48
43. Andrew D Wilson. 2006. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, 255–258.
 44. Robert Zeleznik and Andrew Forsberg. 1999. UniCam - 2D gestural camera controls for 3D environments. *Proceedings of the Symposium on Interactive 3D Graphics*: 169–173.
<https://doi.org/10.1145/300523.300546>
 45. Jamie Zigelbaum, Alan Browning, Daniel Leithinger, Olivier Bau, and Hiroshi Ishii. 2010. g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. In *Proc. TEI*, 261–264. Retrieved September 5, 2010 from <http://portal.acm.org/citation.cfm?id=1709939>
 46. Waymo: Self Driving Cars. Retrieved July 13, 2018 from <https://waymo.com/>
 47. BMW's HoloActive Touch. Retrieved July 13, 2018 from
<https://www.theverge.com/2016/12/15/13965242/bmw-holoactive-touch-virtual-touchscreen-concept>
 48. Leap Motion Sensor. Retrieved from <https://www.leapmotion.com/product/desktop>
 49. Consumer Electronics Show - The Global Stage for Innovation - CES 2019. Retrieved July 13, 2018 from <https://www.ces.tech/>
 50. Infra-red Multi Touch Frames.
 51. Smart Tint ® Smart Film.
 52. Smart Glass Applications with Polymer Dispersed Liquid Crystal (PDLC) Technology | U.S. Green Building Council.

53. BenQ HT2050A 1080P DLP projector.
54. Solid State Relay or Solid State Switch.
55. Arduino Microcontroller.
56. Microsoft Kinect.
57. Play audio and video with MediaPlayer - UWP app developer | Microsoft Docs.
58. Unity 3D Game Engine.
59. Coordinate Systems — Leap Motion C# SDK v3.2 Beta documentation.
60. What is the Bokeh Effect in Photography?
61. Con-Tact Brand Adhesive Removable Chalk Board Covering.
62. Swift Programming Language.
63. 1965 Cadillac DeVille. Retrieved July 13, 2018 from https://upload.wikimedia.org/wikipedia/commons/0/01/1965_Cadillac_Deville_convertible_dashboard1.jpg
64. 1921. Ford Model T Dashboard. Retrieved July 13, 2018 from https://cdn-images-1.medium.com/max/1600/1*RPkqnVho5Q5Q9YSxjtFLJw.png
65. 1948. 1948 Tucker Torpedo. Retrieved July 13, 2018 from <https://azure.barrett-jackson.com/Events/Event/Details/1948-TUCKER-TORPEDO--115982>
66. 1999. Car of the Century? Ford's Model T, of course. Retrieved July 13, 2018 from <https://archive.is/20050204231622/http://info.detnews.com/joyrides/story/index.cfm?id=75#selection-961.0-958.3>
67. 2016. Volkswagen Budd-E. Retrieved July 13, 2018 from <https://media.vw.com/models/budd-e>
68. 2016. BMW AirTouch. Retrieved from

- <https://www.press.bmwgroup.com/global/article/detail/T0247964EN/bmw-group-at-the-ces-2016-in-las-vegas-bmw-presents-the-principle-of-the-contactless-touchscreen-with-airtouch?language=en>
69. 2016. Panasonic's transparent display is hard for your eyes to believe - The Verge. Retrieved July 13, 2018 from <https://www.theverge.com/2016/1/7/10733626/panasonic-transparent-screen-display-ces-2016>
70. 2018. Android Auto. Retrieved July 13, 2018 from <https://www.android.com/auto/faq/>
71. 2018. Apple CarPlay. Retrieved July 13, 2018 from <https://www.apple.com/ios/carplay/>
72. 2018. Google Lens: real-time answers to questions about the world around you. Retrieved from <https://www.blog.google/products/google-lens/google-lens-real-time-answers-questions-about-world-around-you/>
73. 2018. Fakespace Pinch Gloves. Retrieved July 13, 2018 from <https://www.mechdyne.com/hardware.aspx?name=Interaction+Technologies>
74. 2018. iOS - Maps - Apple. Retrieved July 13, 2018 from <https://www.apple.com/ios/maps/>

Appendix

Appendix A: Pre-Pilot Survey

PARTICIPANT #	
GENDER	
AGE	

AUTOPOINT SURVEY (PRE-PILOT TESTING)

1. Using pointing & in-air gesture to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

2. Tapping the window to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

3. Using the Pan gesture to find a time slice of interest felt natural

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

4. Between the two, which did you prefer to use in order to select an area of interest?

Point to select Touch to select

Comment(s):

Other Comments & Suggestions

Appendix B: Pilot Survey

PARTICIPANT #	
GENDER	
AGE	

AUTOPOINT SURVEY (PILOT TESTING)

1. Using pointing & in-air gesture to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

2. Using touch gestures to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

3. Using touch gestures to isolate a target felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

4. Between the two, which did you prefer to use in order to select an area of interest?

Point to select Touch to select

Comment(s):

5. The system can encourage exploration of an unfamiliar area.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

Other Comments & Suggestions

Appendix C: Version 1(V1 survey)

PARTICIPANT #	
GENDER	
AGE	

AUTOPOINT SURVEY (V1)

1. Using pointing & in-air gesture to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

2. Using touch input to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

3. Using touch gestures to isolate a target felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

4. Between the two, which did you prefer to use in order to select an area of interest?

Point to select Touch to select

Comment(s):

5. The system allowed you to isolate a target easily.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

6. The depth-of-field (blurry foreground and background) visualization was useful while isolating a target.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

7. Rank the following techniques (for isolating a target) in order of preference/ usability.

Pan World Tilt Pinch-to-zoom Rotate Depth-of-Field

Comment(s):

8. Suggestions for improving the system (or any other comments).

Appendix D: Version 2 (V2) Survey

PARTICIPANT #	
GENDER	
AGE	

AUTOPOINT SURVEY (V2)

1. Using touch input to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

2. Using point + touch input to select an area of interest felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

3. Using touch gestures to isolate a target felt natural.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

4. Between the two, which did you prefer to use in order to select an area of interest?

Touch to select Point + Touch to select

Comment(s):

5. The system allowed you to isolate a target easily.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

6. Rank the following techniques (for isolating a target) in order of preference/usability.

Pan World Tilt Time Slice

Comment(s):

7. The system can encourage exploration of an unfamiliar area.

1 2 3 4 5

Where 1=strongly disagree and 5 = strongly agree

8. Suggestions for improving the system (or any other comments).



Appendix F: AutoPoint Phase II Questionnaire Data

Participants P0 – P4

Questions	P0	P1	P2	P3	P4
Using touch input to select an area of interest felt natural	5	4	5	5	4
Using point + touch input to select an area of interest felt natural.	2	4	3	4	3
Using touch gestures to isolate a target felt natural	5	2	5	4	4
Between the two, which did you prefer to use in order to select an area of interest (P : Point to Select, T: Touch to Select)	T	T	T	T	T
Comment	Strongly prefer touch only but because point gestures aren't that common	Felt easier to control	The sensor for pointing didn't feel that smooth. There was a lag time that felt less immediate than simply touching.	I think touching to select allowed me to have a more focussed selection of space, but both were cool	
The system allowed you to isolate a target easily.	3	5	5	4	4
Rank the following techniques in order of preference/usability. Pan, World Tilt, Time Slice	1. Pan 2. World Tilt 3. Time Slice	1. Time Slice 2. Pan 3. World Tilt	1. Pan 2. Time Slice 3. World Tilt	1. Time Slice 2. World Tilt 3. Pan	1. Time Slice 2. Pan 3. World Tilt
Comment	Time slice is still very useful, but because I knew the course well enough, I knew when targets would appear		For me, the World Tilt was the least smooth, and most jumpy of all these and I would get confused with the direction of the tilt. I found the time slice direction to be confusing once in a while too.	World Tilt was very cool, but a little finicky. I loved the idea of time slice!	
The system can encourage exploration of an unfamiliar area.	4	5	4	5	4
Other Comments and suggestions for improvements	Maybe zoom-in? Markers for interesting targets , other than the green highlights.	Cool idea	Same direction of pan and world tilt would be nice Smoother user interaction	Maybe the option of being able to zoom out and select another target after you've already selected one	It's probably not possible at this stage, but having there be a video of a natural environment would a better

	Little symbols of interest (like the ones for restaurants /food)			(instead of having to fully exit)	gauge of how natural it truly feels.
--	--	--	--	-----------------------------------	--------------------------------------

Participants P5 – P9

Questions	P5	P6	P7	P8	P9
Using touch input to select an area of interest felt natural	5	5	4	5	3
Using point + touch input to select an area of interest felt natural.	4	4	2	4	3
Using touch gestures to isolate a target felt natural	2	5	4	5	1
Between the two, which did you prefer to use in order to select an area of interest (P : Point to Select, T: Touch to Select)	T	P	T	P	T
Comment	Used to doing this gesture with smart devices.	The ease and lack of tactile engagement made the gesture very appealing. However, the specific gesture felt like it could be strange if done repeatedly	Unsure of point and touch	Touch felt more natural, because it is more similar to existing technology, but Point + Touch was more fun to use and probably more functional.	I felt that I had more control over the touch to select option, whereas the other felt quite unnatural as I'm not used to using advanced technology like that.
The system allowed you to isolate a target easily.	4	4	3	5	3
Rank the following techniques in order of preference/usability. Pan, World Tilt, Time Slice	1. Time Slice 2. Pan 3. World Tilt	1. World Tilt 2. Time Slice 3. Pan	1. Time Slice 2. World Tilt 3. Pan	1. Time Slice 2. World Tilt 3. Pan	1. World Tilt 2. Time Slice 3. Pan
Comment		Being able to shift perspective is the most helpful, as well as engaging to me to get a better look at something going by. Time slice seems very useful, especially if something passed by unexpectedly. Pan is also quite useful, though only marginally compared to the others.	Time Slice is cool idea.	I found that Time Slice was more useful and effective at isolating my target in all situations than Pan was. World Tilt was very useful for target far away or obscured by other objects.	The world tilt allowed me to see targets clear and put them in context of their 360-degree surroundings. Time slice was very useful too in the faster moving car.
The system can encourage exploration of an unfamiliar area.	5	5	5	5	5

Other Comments and suggestions for improvements	Make the world tilt more accessible if possible!	AUDIO Recording	Give option to invert controls for time slicing. Have a virtual cursor floating in space, corresponding to where you are pointing.	Buildings nearer to the center of the screen seemed easier to select. I'm not sure which drag direction for time slice would be more intuitive for people. I got used to it pretty quickly.	It would be interesting to see different environments with each trial because I felt very familiar with the setting after a few rounds of exploring. (i.e. make the simulation itself more complex)
---	--	------------------------	---	---	---

Participants P10 – P14

Questions	P10	P11	P12	P13	P14
Using touch input to select an area of interest felt natural	5	4	5	4	5
Using point + touch input to select an area of interest felt natural.	5	3	3	4	3
Using touch gestures to isolate a target felt natural	3	5	4	4	5
Between the two, which did you prefer to use in order to select an area of interest (P : Point to Select, T: Touch to Select)	P	T	T	P	T
Comment	Maybe just because if was new, but I enjoyed using point + touch	Touch to select feels easier for "real world" (maybe)	The point gesture felt less certain.	Point + Touch felt like you were getting the snapshot perfectly when you wanted to. Touch to select almost felt like more of a hassle and took more effort.	But I guess that's just was I am more used to if I am using a touchscreen.
The system allowed you to isolate a target easily.	4	2	3	4	4
Rank the following techniques in order of preference/usability. Pan, World Tilt, Time Slice	1. World Tilt 2. Pan 3. Time Slice	1. Time Slice 2. World Tilt 3. Pan	1. World Tilt 2. Time Slice 3. Pan	1. World Tilt 2. Time Slice 3. Pan	1. World Tilt 2. Time Slice 3. Pan
Comment	Time Slice was cool, but a bit too slow.	I thought time slice was most "real world" in terms of what I would use . World tilt was also really cool since it allows you to see from an otherwise inaccessible perspective.	For some reason my instinct was to drag the screen opposite of what was intended in the time time slice	I think I needed a view of the entire city (bird's eye view) to select what I wanted to select or wanted to see. Then I wanted to move specifically to the left and right to see my surrounding area that is not visible. Once I did this I panned into an area of choice.	I just liked that the view changed allowing you to see more , and that the buildings were not blocked
The system can encourage exploration of an unfamiliar area.	4	5	5	4	4

<p>Other Comments and suggestions for improvements</p>	<p>I wonder if there's a way to implement this without an opaque screen? It seems counter-intuitive to encourage exploration of an area by blocking the actual area from view.</p>	<p>The different angles seem like they could be nauseating if the movement doesn't correct for that. I would like to be able to "Save" the snapshot/time slice to come back to later to learn more). Maybe synced to my phone's Wikipedia bookmarks and yelp etc. It'd be cool to install this on buses and trains (Greyhound, Amtrak) but not for people to disconnect with the world around them. Probably a delicate balance. Also would have kid-friendly features (or child-lock)</p>	<p>Some option for being able to use left-hand And I'd like to be able to return to a building I had previously looked at.</p>	<p>I think that the touch limited my movement within the city and capabilities of taking a closer look at something that I wanted to see (specifically zooming in was the most difficult part) .</p>	<p>Just better calibration, otherwise great.</p>
--	--	--	--	--	--

Participants P15 – P19

Questions	P15	P16	P17	P18	P19
Using touch input to select an area of interest felt natural	5	5	4	4	4
Using point + touch input to select an area of interest felt natural.	4	5	4	3	3
Using touch gestures to isolate a target felt natural	4	4	4	5	2
Between the two, which did you prefer to use in order to select an area of interest (P : Point to Select, T: Touch to Select)	P	T	P	T	T
Comment	I said 4 for Pointing + T because it's not as natural as touching the windows (it's a learned movement), but it's still extremely easy. I like it better because you don't have to touch the window, which can get it dirty, and someone who's on the middle seat or other side of the car can also do it without having to reach too far.	The time between when I pointed and clicked seemed unnatural as I was gauging what scene I wanted to snapshot, but touching to select the snapshot seemed to present a more specially accurate snapshot of what I wanted to see.	The point + touch technology is really cool.	Felt like I had more control than Pointing	Point felt more unnatural because of the requirement of specific way to do it + have to do it on top of a device.
The system allowed you to isolate a target easily.	5	4	4		4
Rank the following techniques in order of preference/usability. Pan, World Tilt, Time Slice	1. Time Slice 2. World Tilt 3. Pan	1. World Tilt 2. Time Slice 3. Pan	1. Time Slice 2. Pan 3. World Tilt	1. Time Slice 2. World Tilt 3. Pan	1. World Tilt 2. Time Slice 3. Pan
Comment	Time slice is very useful in a real car situation when you are driving past things. World tilt is useful because it allows you to look around the area you are in. Panning is also useful, but it might take some time to find what you want (vs. finding something you just drove past or looking at things from top)	I liked the two-finger world tilt, it felt easy and specially accurate. Time Slice was good because the movement felt natural for if I wanted to "go back in time"	World tilt felt a little unnatural/hard to do		Going back, traveling was smooth.

The system can encourage exploration of an unfamiliar area.	5	5	5	4	5
Other Comments and suggestions for improvements	<p>Honestly impressed and could see it connected to apps that have info about restaurants, for instance. Maybe also connect different cars so you can check road conditions/attractions from cars that drove before you recorded? Integrate screens so people can still play games or other things if they do not want to explore the area? Connect it to phones so people can send/save "screenshots" to friends?</p>	<p>I think this is such a cool idea for people of ALL ages to explore the world. I'd love to hear about your progress., I'll keep an ear out. Suggestions: maybe make height adjustments (I had to being a little to see through the window) and maybe make the car traveling speeds more consistent, add traffic signs etc. :)</p>	<p>This is really exciting!!</p>	<p>Fun interactive, and I'm looking forward to being able to look at trees and other features as targets for more. information</p>	<p>Zooming in to a selected area and being able to explore more interesting areas around would be nice.</p>

Participants P20 – P24

Questions	P20	P21	P22	P23
Using touch input to select an area of interest felt natural	4	3	5	4
Using point + touch input to select an area of interest felt natural.	3	4	4	2
Using touch gestures to isolate a target felt natural	3	3	5	4
Between the two, which did you prefer to use in order to select an area of interest (P : Point to Select, T: Touch to Select)	T	P	T	T
Comment	The point + touch was not a gesture I would naturally do. Touching the window seemed more natural with less effort.			
The system allowed you to isolate a target easily.	3	2	5	3
Rank the following techniques in order of preference/usability. Pan, World Tilt, Time Slice	<ol style="list-style-type: none"> 1. Time Slice 2. Pan 3. World Tilt 	<ol style="list-style-type: none"> 1. Time Slice 2. World Tilt 3. Pan 	<ol style="list-style-type: none"> 1. Pan 2. Time Slice 3. World Tilt 	<ol style="list-style-type: none"> 1. Time Slice 2. World Tilt 3. Pan
Comment	The system for tilting and zooming did not always respond reliably (like an iphone)		Pan was the gesture I felt most used to because it was similar to what I use daily	Time Slice is neat. When I was young, on multiple occasions I wanted my sister to see something out of my side of the window but by the time I got her attention, it was too late. This would have been sooooo useful
The system can encourage exploration of an unfamiliar area.	5	4	4	5
Other Comments and suggestions for improvements	Maybe voice-commands. Being able to see real world in real time while interacting with a snapshot (small window in corner)	Be cautious about potential effects on young child development, so much screen time.		I would love to see a game like Pokemon Go on the car window! Using world tilt to find a Pokemon would be fun